

SVEUČILIŠTE U ZAGREBU  
FILOZOFSKI FAKULTET

MARKO TADIĆ

RAČUNALNA OBRADA MORFOLOGIJE  
HRVATSKOGA KNJIŽEVNOG JEZIKA

DOKTORSKA DISERTACIJA

ZAGREB, 1994.

*Posvećeno trima »krivcima« za odabir lingvistike kao moga životnog poziva: mome djedu Dušanu Jakcu koji me prvi upućivao u stvari duha, Josipu Tabaku koji mi je prvi približio blago jezika hrvatskoga i Tonku Maroeviću koji mi je etimologiju predstavio u liku kriminalističkoga romana.*

*Zahvaljujem se svojim roditeljima, supruzi i svim bližnjima na strpljenju bez kojega ne bih nikada uspio privesti kraju ovaj rad.*

*Konačno, ali ne i manje važno, zahvaljujem svome mentoru prof. dr. Milanu Mogušu na mnogim plodnim diskusijama u kojima je sazio velik dio ovdje izloženih problema.*

# KAZALO

<b>1. UVOD .....</b>	<b>6</b>
<b>2. CILJ RADA I GRANICE PROBLEMA.....</b>	<b>8</b>
<b>2.1. Cilj.....</b>	<b>9</b>
<b>2.2. Ograničenje problema .....</b>	<b>12</b>
2.2.1. Morfologija nasuprot sintaksi .....	12
2.2.2. Fleksija nasuprot derivaciji .....	16
2.2.2.1. Glagolski vid .....	18
2.2.2.2. Drugi glagolski oblici .....	19
2.2.2.3. Kategorija lica u zamjenica.....	20
2.2.2.4. Komparativ i superlativ .....	21
2.2.2.5. Problem tvorbe priloga od pridjeva.....	25
2.2.2.6. Ostala ograničenja u fleksiji .....	26
2.2.3. Promjenljivost nasuprot nepromjenljivosti .....	27
2.2.3.1. Promjenljivost priloga.....	27
2.2.3.2. Promjenljivost brojeva .....	28
2.2.3.3. Prijedlozi.....	30
2.2.4. Sinteza nasuprot analizi.....	31
2.2.5. Realizacija postava pismom.....	31
2.2.5.1. Pisanje dž, lj, nj, št, žd.....	32
2.2.5.2. Pisanje jata .....	33
2.2.5.3. Stanje pravopisa .....	34
<b>3. TEORIJSKE POSTAVKE.....</b>	<b>36</b>
<b>3.1. Problemi klasifikacije vrsta riječi u hrvatskome .....</b>	<b>41</b>
<b>3.2. Dva postulirana konstrukta .....</b>	<b>46</b>
3.2.1. Vrste promjene.....	48
3.2.1.1. Nulta promjena .....	48

3.2.1.2. Deklinacija .....	49
3.2.1.3. Konjugacija .....	51
3.2.1.4. Komparacija.....	52
3.2.2. Uzorci promjene.....	53
3.2.2.1. Deklinacija .....	53
3.2.2.2. Konjugacija .....	54
3.2.2.3. Komparacija.....	57
<b>3.3. Odnosi tipova promjene prema tradicionalnim vrstama riječi .....</b>	<b>57</b>
<b>3.4. Razine obrade .....</b>	<b>59</b>
3.4.1. Dvije razine obrade .....	59
3.4.2. Postave na dvije razine.....	62
<b>4. KLASIFIKACIJA UZORAKA PROMJENE .....</b>	<b>66</b>
<b>4.1. Nulta promjena.....</b>	<b>66</b>
<b>4.2. Deklinacija.....</b>	<b>67</b>
4.2.1. a-vrsta.....	67
4.2.2. e-vrsta .....	79
4.2.3. i-vrsta.....	80
4.2.4. Poimeničeni pridjevi .....	81
4.2.5. Pridjevska deklinacija.....	81
4.2.5.1. Neodređeni i određeni oblik .....	81
4.2.5.2. Zamjenička deklinacija.....	82
<b>4.3. Konjugacija .....</b>	<b>83</b>
4.3.1. Neplodni glagoli .....	83
4.3.2. Plodni glagoli.....	86
<b>4.4. Komparacija .....</b>	<b>88</b>
4.4.1. Komparativ .....	88
4.4.2. Superlativ .....	88
<b>5. RAČUNALNI MODEL GENERIRANJA OBLIKA .....</b>	<b>89</b>
<b>5.1. Definicija i organizacija podataka .....</b>	<b>89</b>
5.1.1. Nastavci.....	90
5.1.2. Preoblike .....	98
5.1.2.1. Temeljna struktura preoblika .....	98
5.1.2.2. Opis preoblika .....	101

5.1.3. Leksikon .....	114
5.1.3.1. Uloga leksikona u modelu.....	114
5.1.3.2. Struktura leksikonskoga ulaza.....	116
<b>5.2. Programi.....</b>	<b>120</b>
5.2.1. Princip rada.....	120
5.2.2. Izvorni kodovi programa .....	121
5.2.2.1. GENOBLIK .....	121
5.2.2.2. GENIMEN .....	122
5.2.2.3. GENPRID .....	128
5.2.2.4. GENKOMP .....	134
5.2.2.5. GENPRIL .....	138
5.2.2.6. GENGLAG.....	141
<b>6. ZAKLJUČAK I PERSPEKTIVE .....</b>	<b>148</b>
<b>7. LITERATURA .....</b>	<b>151</b>
7.1. Knjige i članci.....	151
7.2. Rječnici.....	160

# 1. UVOD

Jedna od najuočljivijih osobina jezične razine hrvatskoga književnog jezika kojom se bavi ovaj rad jest složenost. No tim je prije izazov, koji njezin opis predstavlja, veći. Ovaj rad iznosi pokušaj — ne prvi, ne jedini a zacijelo ne i najbolji — da se provede opis morfologije hrvatskoga književnog jezika te se taj lingvistički opis potkrijepi računalnim modelom. Naime, valjalo bi odmah na početku naglasiti da je za razliku od strogo informatičkoga pristupa prirodnome jeziku kao skupu specifičnih podataka, koji pristup teži što učinkovitijem obliku njihove obrade, temeljni pristup ovoga rada zapravo lingvistički. Ovdje je prvenstveno riječ o uporabi računala u lingvističke svrhe tj. o uporabi računala u lingvistici i to za potrebe testiranja lingvističkoga modela jednoga od jezičnih podsustava. U ovom je to slučaju morfološki podsustav hrvatskoga književnog jezika. Stoga je ishodišno znanstveno područje ovoga rada svakako lingvistika. Konačno, i sâm je problem »načet« upravo s te strane. Računalna realizacija lingvističkoga modela služi ponajprije za provjeru samoga modela i njezina izvedba nipošto nije lišena nedostataka koje bi informatičar mogao uočiti već na prvi pogled. Nema sumnje da se upravo u tom računalnoizvedbenom dijelu rada može pronaći dovoljno prostora za naknadne dorade. Time bi se zacijelo dobilo na brzini odvijanja programa ili bi se kompresijom podataka mogle reducirati memorijske potrebe ili bi se dodatnim kodiranjem kombinacija nastavaka i(li) preoblika moglo doći do sažetijega zapisa. No temeljno bi načelo funkcioniranja sustava ostalo nepromijenjeno. Osim toga, čini se da je model ovako izložen unekoliko pregledniji.

Sámo rješenje o kojem je ovdje riječ, a koje rješenje uključuje i lingvistički model i njegovu računalnu izvedbu, obilježavat će se nadalje kraticom tj. imenom sâmoga računalnog programa — GENOBLIK. Pod tim će se imenom podrazumijevati i lingvistički model i njegova računalna izvedba. Nastojat će se što je više moguće da iz samoga konteksta uporabe termina bude jasno o kojem se vidu toga sustava govori.

Sam je rad sastavljen od sedam dijelova. Nakon uvoda slijedi drugi dio kojim se iznosi cilj rada i pokušavaju odrediti granice problema. U trećem se dijelu iznose teorijske postavke za organizaciju lingvističkoga modela, a u četvrtom se iznosi sâm lingvistički model u obliku klasifikacije uzoraka promjene. Peti dio govori o samoj obradi građe, organizaciji i naravi podataka te donosi izvorne kodove kojima je realiziran računalni model. Nakon šestoga, zaključnoga dijela, slijedi popis literature.

## 2. CILJ RADA I GRANICE PROBLEMA

Dosada su poznati različiti računalni sustavi<sup>1</sup> koji su se okušali u obradi morfologije pojedinoga prirodnog jezika. Dio njih je, s jedne strane, donekle prisiljen ograničenjima računalnotehničke prirode, nerijetko posezao za rješenjima koja nisu uvijek u potpunosti poštivala jezične zakonitosti, barem ne u onom obliku u kojem ih iskazuju gramatike. S druge se strane, dio njih, unatoč lingvistički relevantnim postavkama, služio metodama za koje se u lingvistici ne bi moglo naći opravdanja.<sup>2</sup> Takvi su pristupi često rezultirali nepotpunim »pokrivanjem« pojava u morfologiji.

Stoga je polazna zamisao ovoga rada bila na tragu onoga što je dijelom objavljeno u Tadić (1992). Jedan je od temeljnih zahtjeva izvedba računalnoga sustava koji bi u najvećoj mogućoj mjeri poštivao propis što ga na razini morfologije daje gramatika hrvatskoga jezika. S razlogom se može pretpostaviti da bi s pristupom koji dosljedno vodi računa o jezičnim činjenicama (prije svega segmentaciji jezičnih jedinica), rizik stvaranja empirijski neadekvatne teorije<sup>3</sup> morao biti umanjen. No pravu će provjeru GENOBLIK imati tek kada ga se iskuša na opsežnijem korpusu

---

<sup>1</sup> Pregled značajnijih dosadašnjih bavljenja problemom iznosi se u Tadić (1992) u 3. poglavlju.

<sup>2</sup> v. Tadić (1992:5).

<sup>3</sup> Empirijski neadekvatne teorije (Chomsky 1965:61) nemaju dovoljnu *slabu generativnu sposobnost* za proizvođenje svih ovjerenih postava a time, dakako, ni potpuna opisa prirodnoga jezika te takve teorije ne prolaze empirijski test adekvatnosti.



hrvatskoga jezika u kojemu će se naći dovoljan broj traženih potvrda pojedinih oblika neke riječi kojih oblika u rječnicima najčešće i nema.

## 2.1. Cilj

Sâm je cilj rada zapravo dvojake naravi:

1. izraditi lingvistički opis promjene oblika riječi u hrvatskome književnom jeziku;
2. taj opis primijeniti u računalnom sustavu koji bi bio kadar generirati sve moguće oblike riječi.

Temelj za lingvistički opis dijelom je izložen u tradicionalnom taksonomskom gramatičkom obliku u Babić-Brozović-Moguš-Pavešić-Škarić-Težak: *Povijesni pregled, glasovi i oblici hrvatskoga književnog jezika*.<sup>4</sup> Tako uzeta osnovica dorađena je i proširena, a opis je uvođenjem dvaju teorijskih konstrukata sa statusom postulata<sup>5</sup> istodobno pojednostavnjen. Jedan od razloga za proširenje postojećega opisa zahtjev je koji je postavljen od samoga početka, a to je: model mora biti kadar opisati *sve moguće oblike* neke riječi bez obzira na njihovu učestalost i očekivanost u svakodnevnoj jezičnoj porabi. To često uključuje i oblike koje bi mnogi izvorni govornici hrvatskoga danas označili kao izrazito stilski obilježene. No takav zahtjev ima svoja metodološka i teorijska opravdanja dijelom u generativnoj gramatici,<sup>6</sup> a dijelom u historijskoj gramatici

---

<sup>4</sup> Pavešić-Težak-Babić: »Oblici hrvatskoga književnog jezika (morfologija)« u Babić-Brozović-Moguš-Pavešić-Škarić-Težak: *Povijesni pregled, glasovi i oblici hrvatskoga književnog jezika*, HAZU-Globus, Zagreb 1991, str. 473-741. Nadalje će se na ovu publikaciju referirati kraticom GRAM.

<sup>5</sup> v. поближе u 3.2.

<sup>6</sup> v. поближе u 3.

hrvatskoga jezika.<sup>7</sup> Ovaj se rad bavi isključivo promjenom oblika riječi te je stoga usredotočen samo na tu razinu promatranja pokušavajući usustaviti i eksplicirati sve načine na koje se flektivno promjenljive riječi u hrvatskome mogu oblikovati.

Drugi razlog za proširenje postojećega tradicionalnogramatičkoga opisa nametnuo se upravo s potrebom za računalnom realizacijom. Mnogi su elementi opisa u GRAM postavljeni fakultativno. Oprimjerimo to opisom samo jedne specifičnosti imenične deklinacije gdje formulacije poput:

...To [proširenje s umetkom *-ov-/-ev-*] biva u jednosložnih osnova (*s nekim izuzecima*) i *nekih* dvosložnih, ali *većina njih može imati* i kratku množinu, neke *gotovo jednako često*, neke *rjeđe*, neke *samo u stilski obilježenom tekstu*... (t. 146)

...Imenice koje u Njd imaju jedan slog *gotovo sve* obilježavaju Nmn *i* umetkom *-ov-/-ev-* ispred nastavka *-i*... (t. 151)

...Gotovo sve te imenice *mogu* imati i oblik bez umetka, ali se takvi oblici *većinom rijetko ili rjeđe upotrebljavaju* te imaju stilističku vrijednost... (t. 152)

...*Običniji* je dugi oblik u imenica (...) Kratki oblik tih imenica ima stilističku vrijednost. (t. 154)<sup>8</sup>

nisu rijetkost. Takav je opis gotovo nemoguće pretvoriti u računalni model jer se ne navodi koje sve imenice tvore množinu s umetkom (ako se kakav

---

<sup>7</sup> Valja, međutim, napomenuti da se ovaj rad ograničuje isključivo na sinkronijsku razinu. Dakako da se može postaviti pitanje o prirodi sinkronije no to bi nas odvelo daleko van okvira ovoga rada. Sinkronijska se razina promatranja ovdje preuzima kao gotov metodološki obrazac u kojem se istraživanje obazire samo na jezične pojave koje postoje u istom vremenskom presjeku.

<sup>8</sup> GRAM, str. 508-511. (isticanja moja).

popis i daje, on obično završava s tri točke). Naime, računala su kadra obraditi samo ono što im se zada izrijekom, a u takvim je formulacijama previše toga ostavljeno u implicitnom obliku. Kako računala barataju prvenstveno s entitetima diskretne prirode, ovakve su se formulacije morale pretvoriti u nedvosmislene eksplicitne tvrdnje iz kojih se za svaku pojedinu imenicu vidi tvori li množinu s umetkom ili ne. Ne čini se preuzetnim očekivati da je takvo proširenje ujedno dovelo i do poboljšanja postojećega opisa jer je postao obuhvatniji i dosljedniji upravo zbog s računalne strane nametnutog zahtjeva za eksplicitnošću. Za sam je lingvistički opis dobro da je iscrpan: ako već propisom nije moguće obuhvatiti sve slučajeve onda dodatni popis jezičnih jedinica *mora* biti iscrpan jer time stječe ulogu propisa. Dakako, da je s druge strane jedan od zahtjeva bio održati leksikon što je moguće manjim tj. zapis u njemu što je moguće sažetijim ravnajući se prema tome da je propisom potrebno obuhvatiti što više pravilnosti kako popis ipak ne bi isuviše narastao.

Gornji je primjer samo ilustracija vrste prepreka na koje se nailazi pri pokušaju računalne obrade hrvatskoga jezika. Bit je problema u tome što, unatoč višestoljetnoj slovničkoj tradiciji, nemamo niti jednoga dovoljno formalnoga (ili suvremenoga) lingvističkoga opisa hrvatskoga jezika u cjelosti koji bi mogao poslužiti kao dobra podloga za izvedbu računalnoga modela.<sup>9</sup> Utoliko je jedan od ciljeva ovoga rada opis morfologije hrvatskoga u obliku dovoljno formaliziranome da se na njegovoj osnovi može izgraditi

---

<sup>9</sup> Jednu od iznimaka čini knjiga Zrinke Babić *Generativni opis konjugacijskih oblika*, HFD, Zagreb 1991. koja je u mnogo čemu bila korisna i nadasve inspirativna.

računalni model s dvije razine<sup>10</sup> — ulaznom ili morfotaktičkom i izlaznom ili razinom oblika riječi.<sup>11</sup> S tim je osnovnim ciljem sâm opis i rađen.

## **2.2. Ograničenje problema**

Potrebno je odmah na početku ograničiti područje kojim se ovaj rad bavi i pokušati definirati opsege osnovnih termina.

### **2.2.1. Morfologija nasuprot sintaksi**

Tradicionalna gramatika rabi termine riječ i rečenica bez većih problema. Međutim nakon de Saussureova koncepta minimalnoga jezičnog znaka i strukturalističkoga poimanja jezika, termin *riječ* postaje s lingvističkoga stajališta problematičan zbog svoje mnogoslojnosti i nikad postignute dovoljne preciznosti u njegovu određivanju. Pa ipak, riječ (ili bolje rečeno njezini oblici) čini osnovni predmet ovoga opisa. U temelju odnosa između morfologije i sintakse stoji problem delimitiranja jezičnih razina opisa i pripadajućih im jedinica. Riječ je tu u žarištu interesa s koje god strane gledali.

Slijedeći stroge distribucionalističke zahtjeve (za definicijom jezičnih jedinica postupkom komutacije na višim razinama) Chomsky je u prvoj verziji svoje teorije jezika tj. gramatike isključio ne samo postojanje riječi i

---

<sup>10</sup> v. Tadić (1992b:passim).

<sup>11</sup> U Babić (1991), str. 27 unutar teorijskoga okvira generativne fonologije predlažu se hrvatski termini za te dvije razine: 1. apstraktna ili ishodišna tj. fonološkosustavna; 2. izvedena ili konkretna tj. fonetsko-sustavna.

morfološke sastavnice gramatike, nego i postojanje samostalnog leksikona.<sup>12</sup> U tako zamišljenoj teoriji jezika morfološka se problematika rješavala isključivo sintaktičkim sredstvima. Na svu sreću, takav je radikalan stav doživio promjene i već u Chomsky (1965) leksikonu se dodjeljuje status samostalne gramatičke sastavnice što je kasnije pružilo mogućnosti za istraživanja zakonitosti leksikona neovisnih od sintaktičke komponente. Sâm Chomsky 1970. u članku »Remarks on Nominalizations« uvodi *leksikalističku hipotezu* prema kojoj se sintaksa niti bavi, niti ima pristupa unutarnjem obliku riječi.<sup>13</sup> »To je automatski značilo da leksikon više nije bio promatran kao skup morfema, već kao skup riječi i da tvorba riječi više nije dio sintakse.«<sup>14</sup> Kasniji radovi uvode sve više elemenata na kojima se može zasnivati zasebna morfološka sastavnica gramatike.<sup>15</sup> Aronoff u morfološku komponentu (koja još uvijek prvenstveno obuhvaća derivaciju) generativne gramatike vraća riječ kao minimalnu jedinicu<sup>16</sup> tvrdnjom:

All regular word-formation processes are word-based. A new word is formed by applying a regular rule to a single already existing word.<sup>17</sup>

Time se izravno uvod i potreba za razinom promatranja koja više ne uzimlje u obzir samo kombinacije morfema ( $M_1 + M_2$ ) već i kombinacije tih

---

<sup>12</sup> v. Chomsky (1957:42-43), v. Hammond-Noonan (1988:1-10).

<sup>13</sup> v. Anderson (1988:23).

<sup>14</sup> Mihaljević (1991:81).

<sup>15</sup> Halle (1973) prvi daje nacrt zasebne morfološke komponente generativne gramatike koja je komponenta zadužena za generiranje oblika riječi.

<sup>16</sup> v. i Hammond-Noonan (1988:5) i Mihaljević (1991:82). Taj je stav dobio naziv »word-based morphology«.

<sup>17</sup> Aronoff (1976:21): Svi su pravilni procesi tvorbe riječi temeljeni na riječima. Nove se riječi tvore primjenom pravila na već postojeću pojedinačnu riječ. (prijevod moj).

kombinacija  $((M_1 + M_2) + M_3)$  gdje su neke kombinacije  $(M_1 + M_2)$  već gotove riječi koje postoje u leksikonu. Među novijim radovima koji se bave morfološkom problematikom (i definicijama riječi)<sup>18</sup> prednjači rad Di Sciullo-Williams (1987). U njemu se afirmira postojanje riječi kao jezične jedinice i iznose se tri temeljne ideje koje stoje iza toga termina:

The notion central to this text concerns the *form* of a certain set of objects; the definition of this set is the theory of morphological form. The theory defines the set by means of a set of atoms (morphemes) and rules of combination (affixing, compounding). We will call the members of this set *morphological objects* and assert that the central task of morphology is to determine the laws of form that determine membership in this set. Just as morphology has atoms, so does syntax, and words are commonly taken to be the atoms of syntax. We will call words in this sense *syntactic atoms*. This notion of word is conceptually and factually distinct from that of word as »morphological object«. (...) Finally, the third notion of word (...) For this notion of word, which we believe to be of no interest to the grammarian (though perhaps it is to the psychologist), we coin the term *listeme*. We will show that the listemes of a language correspond to neither the morphological objects nor the syntactic atoms of a language.<sup>19</sup>

---

<sup>18</sup> Anderson (1988), Carstairs (1988), Perlmutter (1988), Stemberger-MacWhinney (1988), Bybee (1988), Dressler (1988).

<sup>19</sup> DiSciullo-Williams (1987:1): Središnji pojam ovoga teksta obuhvaća *formu* stanovitoga skupa objekata; definicija toga skupa predstavlja teoriju morfološke forme. Teorija definira skup putem skupa atoma (morfemi) i pravila njihova kombiniranja (afiksacija, slaganje). Članove toga skupa zvat ćemo *morfološkim objektima* i ustvrditi da je osnovni cilj morfologije odrediti zakone forme koja određuje pripadnost tome skupu. Baš kao što morfologija ima atome, ima ih i sintaksa a riječi su obično te koje se smatraju njenim atomima. *Sintaktičkim* ćemo *atomima* zvati riječi u tome smislu. Ovakva je koncepcija riječi pojmovno i činjenično odjelita od one koja riječ gleda kao »morfološki objekt« (...) »

S obzirom da je cilj ovoga rada opis promjene oblika riječi, prvi stav iz ovoga citata pruža traženu teorijsku platformu s koje se može krenuti dalje. Naime, sintaktički atomi nisu kombinacije minimalnih jezičnih znakova (morfema) već su to riječi ili čak skupine riječi (npr. takvi su sintaktički atomi u hrvatskome složeni glagolski oblici ili neodređene zamjenice tvorene sa *ni*-<sup>20</sup>).

...the theory of grammar has two subtheories, morphology and syntax, each with its own atoms, rules of formation (...). Morphology and syntax are different (though similar) sciences about different objects, so the idea that the derivations in one could get mixed up with those of the other should not arise in the first place.<sup>21</sup>

---

Konačno, treća ideja riječi (...) Za tu smo ideju riječi, za koju vjerujemo da gramatičarima nije zanimljiva (premda bi mogla biti psiholozima), iskovali termin *listem*. Pokazat ćemo da se listemi jezika razlikuju i od morfoloških objekata i od sintaktičkih atoma jezika. (prijevod moj).

<sup>20</sup> Mogao bi se ovdje uputiti prigovor navođenjem složenih oblika riječi (ili oblika s klitikama) koji su zapravo granični slučajevi. To nije toliko vidljivo pri proizvođenju oblika koliko je uočljivo pri analizi gdje je čest slučaj da se npr. u hrvatskome pomoćni glagol složenoga glagolskog vremena (*Ona je često za njim pogledavala.*) ili neodređena čestica u razdvojenim neodređenim zamjenicama (*Ni u kojem slučaju nemojte to učiniti!*) nalazi na udaljenom mjestu u sintaktičkoj strukturi. No ta je udaljenost prisutna tek u površinskoj strukturi jer je zacijelo rezultat fakultativne transformacije kojom se odlikuje visoki stil. To ipak ostaju granični slučajevi: i takva fakultativna transformacija pri analizi oblika riječi u nekoj rečenici već znači, premda djelomično, uvlačenje sintakse u analizu. Aronoff (1976:3) navodi slične pojave u klasičnom hebrejskom i navaškom za koje predlaže istu metodu razrješenja — transformaciju premještanja.

<sup>21</sup> DiSciullo-Williams (1987:46): ...teorija gramatike sastoji se od dvije podteorije, morfologije i sintakse, svake sa svojim atomima, pravilima oblikovanja (...). Morfologija i sintaksa su različite (premda slične) discipline o različitim objektima, te nikako ne bi na prvo mjesto valjalo smještati pomisao da se derivacije iz jedne mogu miješati s derivacijama iz druge. (prijevod moj)

Ono što se ovim radom pokušava uspostaviti je jedan od mogućih popisa pravila kombinacija morfoloških entiteta (ili tradicionalnije rečeno: osnova i nastavaka) koje kombinacije čine različite oblike riječi u hrvatskome. Najjednostavnije rečeno sintaktička komponenta gramatike tim pravilima najčešće ne vlada već joj ulaz čine gotovi oblici riječi s pratećim kategorijalnim oznakama kojima se regulira njihovo uvrštavanje u rečeničnu strukturu.<sup>22</sup> Morfološka je sastavnica gramatike zadužena za generiranje tih oblika. Ovaj je pokušaj, koji želi opisati morfološku sastavnicu i modelirati njezinu generativnu sposobnost, zapravo opis »iznutra« koji se ne obazire na kasniju službu tih riječi u rečenici već je usredotočen na što gospodarniji opis tipova promjene oblika riječi.

Ovdje se pokušava dati klasifikacija riječi s obzirom na *vrstu promjene* njezinih oblika i *uzorak promjene* tih oblika. Taj se pristup, premda taksonomski u svojoj naravi, ipak ne obazire na tradicionalnu podjelu riječi na vrste već je s njom korespondira na način prikazan u **3.3**.

### 2.2.2. Fleksija nasuprot derivaciji

Kada se govori o morfologiji valja imati na umu opseg termina. Morfologija u užem smislu obuhvaća samo fleksiju, a u širem i fleksiju i derivaciju.<sup>23</sup>

Morfološkom se komponentom u generativnoj gramatici bave dvije struje teoretičara koje su gotovo najviše suprotstavljene upravo oko pitanja razlikovanja fleksije od derivacije. To su *teorije razdvojene morfologije* (split morphology theories) i *teorije leksičke morfologije* (lexical

---

<sup>22</sup> v. Chomsky (1965:171), Anderson (1988:25-27).

<sup>23</sup> v. Babić (1986), str. 12, t. 27.



morphology theories).<sup>24</sup> Prvi modeli,<sup>25</sup> flektivnu morfologiju smještaju izvan leksikona i smatraju je sastavnicom s vlastitim pravilima odjelitom od one koja određuje derivacijske procese. Drugi<sup>26</sup> se, naprotiv, zalažu za jedinstvenu gramatičku komponentu koja objedinjuje sve procese oblikovanja riječi, ali u tom slučaju unutar te komponente postoji potreba za razlikovanjem razina na kojima se pravila primjenjuju. Ta pravila moraju biti ciklična i(li) neciklična, te su uz njih potrebni i blokatori cikličnosti. Argumenti za prvu skupinu modela su otprilike slijedeći:<sup>27</sup>

- fleksija u većini jezika svijeta nastupa nakon što je derivacija završila svoj posao (oblikovala osnovu);
- flektivni se sustavi u najčešćem broju slučajeva služe gramatičkim morfemima kojih plan sadržaja (jedan morf) na planu sadržaja istovremeno ostvaruje više semova (gramatičkih kategorija) dok za takvu pojavu u derivacijskih morfema gotovo da i nema potvrde;<sup>28</sup>

---

<sup>24</sup> de Bleser-Bayer (1988:45).

<sup>25</sup> npr. Aronoff (1976), Anderson (1982).

<sup>26</sup> O drugoj struji vidi potanko u Mihaljević (1991:81-87).

<sup>27</sup> Anderson (1988:30, 28 i 29). Argumenti za drugu skupinu modela mogu se naći u Mihaljević (1991:85-87).

<sup>28</sup> Prihvati li se takva tvrdnja aglutinacija se, dakako, mora smatrati derivacijom, a nikako fleksijom jer u protivnom tvrdnja ne stoji. Stoga bi precizniji termin bio »fuzijski sustavi«. Rasprava, premda nesumnjivo zanimljivu, o fuzijskim i aglutinativnim sustavima u svjetlu razlike između derivacije i fleksije zagazila bi u područje daleko izvan domene ovoga rada stoga bi je valjalo ostaviti za neku bolju priliku.

- čini se da ne postoje morfemi u kojima bi se moglo uočiti miješanje flektivnih s derivacijskim kategorijama.<sup>29</sup>

Budući da je postavljen bio zahtjev za opisom fleksije, modeli prve skupine pružaju zadovoljavajuću teorijsku osnovicu za nastavak istraživanja.

Dosadašnji su računalno realizirani sustavi pokazivali različite sposobnosti u tretiranju ovih dvaju područja: neki su pokrivali obje, neki samo fleksiju, a neki samo derivaciju. GENOBLIK ne bavi se derivacijom (premda nije isključeno da se princip njegova funkcioniranja ne bi mogao primijeniti i na tvorbu riječi, no to je tema koja bi zacijelo zahtijevala i redefiniciju teorijskoga pristupa<sup>30</sup>) nego samo fleksijom. Međutim, kao drugdje i ovdje postoje granični slučajevi.

#### 2.2.2.1. *Glagolski vid*

Kako tretirati glagolski vid? Treba li ga uključiti u fleksiju? Argument za bila bi pravilnost pojavljivanja u paru glagola s oprekom svršenost/nesvršenost kao i malen inventar sufikasa kojima se glagoli perfektiviziraju. Nadalje, tu je i gospodarnost (p)opisa ako se oblici vidske promjene mogu prikazati pod istom natuknicom u leksikonu.<sup>31</sup> Ali:

Perfektivizacija se postiže prefiksalmom i sufiksalmom *tvorbom*, (sic!)  
(...) Parovi nastali sufiksacijom čine bližu vidsku opreku jer obično

---

<sup>29</sup> Miješanjem se ne bi trebali smatrati slučajevi kad se gotovi derivirani oblici iskoriste u oblikovanju složenih oblika paradigme kao u npr. hrvatskih glagola.

<sup>30</sup> v. Tadić (1992:7) gdje se prenosi zanimljiv pogled iz Di Sciullo-Williams (1987:69-71) koji unutar teorije o riječi kao morfološkom objektu izjednačuje fleksiju i derivaciju na funkcionalnoj razini što bi moglo biti relevantno za eventualni pokušaj primjene principa iz sustava poput GENOBLIK-a na derivacijsku morfologiju.

<sup>31</sup> v. Becker Makkai (1969:159-160).

imaju istovjetno leksičko značenje. Parovi tvoreni prefiksacijom mogu imati isto leksičko značenje, ali najčešće imaju i neke značenjske razlike. Kako se glagoli prefektiviziraju i imperfektiviziraju, kazuje se u poglavlju o tvorbi glagola u *Tvorbi riječi*. (sic!)<sup>32</sup>

U hrvatskom književnom jeziku (...) gramatička oznaka vida obično se veže za glagol tako da svaki ima svoj vid, pa je izbor gramatičke oznake vida ujedno i *leksički izbor* odgovarajućega glagola. Neki su glagoli dvovidski, pa se upotrebljavaju sad u jednom, sad u drugom vidu, a da to nikako nije morfološki označeno. Čitavo to pitanje pripada području glagolske **tvorbe** ili leksikologije.<sup>33</sup>

Rješenje koje je ovdje prihvaćeno kreće od toga da su vidske promjene zapravo pojava iz domene derivacijske morfologije te je vidskim parnjacima stoga mjesto u leksikonu kao zasebnim leksikonskim ulazima. No to ne znači da je kategorija vida ispuštena iz opisa. Naprotiv, ona je i te kako potrebna na apstraktnoj razini. Njena se realizacija vidi u postojanju odabira između imperfekta i aorista glagola u različitim uzorcima glagolske promjene.

#### 2.2.2.2. *Drugi glagolski oblici*

Promatrajući u hrvatskome samo načine promjene oblika riječi unutar konjugacije može se primijetiti da ona predstavlja vrstu promjene bitno različitu od ostalih, prije svega po svojoj složenosti. Glagolska paradigma tj. konjugacija određena je većim brojem kategorija nego ikoja druga, međutim, ono što je na ovome mjestu interesantno načini su na koje se

---

<sup>32</sup> GRAM 669-670. (isticanja moja).

<sup>33</sup> Katičić (1986:43) (isticanja moja).

glagol oblikuje unutar infinitnoga dijela paradigme. S pravom bi se moglo pitati gdje je točna granica fleksije i derivacije u tom slučaju kad dijelove glagolske paradigme čine dva priloga (glagolska) i dva pridjeva (glagolska). No GENOBLIK pridržava se tradicionalnoga rješenja i zadržava postojeće gledanje na glagolsku paradigmu te tako derivacijske procese *tvorbe* glagolskih priloga i pridjeva po svojim postupcima karakteristične za derivaciju, uključuje u fleksiju.<sup>34</sup>

No uz složene glagolske oblike vezano je još jedno pitanje. Što točno znači zahtjev za proizvođenjem *svih oblika*? Ovaj pristup, u svojoj orijentaciji na praktičnost, pokušava se usredotočiti na one oblike koji su doista imaju različite likove, koji pokazuju razno-likost te nastoji izmaknuti zalihosnim ponavljanjima kakva se znaju pojaviti unutar npr. glagolske paradigme. To je jedan od osnovnih razloga zašto se od glagolskih oblika ne obrađuju oni složeni — njihov se opis može izvesti spajanjem pravila za oblikovanje glagolskoga pridjeva bilo kojega glagola i pravila za oblikovanje vremena i načina pomoćnih glagola.<sup>35</sup>

Još je jedna kategorija vezana uz glagole koja se ne obuhvaća eksplicitno sustavom GENOBLIK, a to su nepotpuni i bezlični glagoli. Oni se ionako upotrebljavaju samo u trećem licu i gotovo da ne podliježu promjeni. Stoga je za takve glagolske oblike predviđeno rješenje koje ih tretira kao riječi bez promjene.

---

<sup>34</sup> U daljnjim bi istraživanjima možda moglo biti zanimljivo razmotriti mogućnost dekonstrukcije takve paradigme prema strožim kriterijima razdiobe fleksije i derivacije te raspoređivanje njezinih elemenata po drugim bliskim sastavnicama gramatike. To bi vjerojatno bilo moguće ako bi se vrlo precizno (p)opisala uloga tih dijelova glagolske paradigme na sintaktičkoj razini tj. u sintaktičkoj strukturi.

<sup>35</sup> v. dolje **3.4.**

### 2.2.2.3. Kategorija lica u zamjenica

Promotrimo slijedeću tvrdnju iz GRAM:

Kategorija lica izražena je u ličnim zamjenica i glagola, a posredno i u nekih zamjenica i priloga.<sup>36</sup>

Tako postavljena tvrdnja nesumnjivo bi morala utjecati na organiziranje paradigme zamjenica, koja bi paradigma morala osim padeža, broja i roda, uključiti i četvrtu dimenziju tj. oblike lica. Međutim, za razliku od pridjevske paradigme kod koje četvrta dimenzija tj. određenost ima svega dvije vrijednosti (određeni/neodređeni pridjev), kod zamjenica ona ima tri (1, 2. i 3. lice). K tome još zamjenice obiluju i s više dodatnih, supletivnih, najčešće klitičkih oblika. Tako zamišljena paradigma morala bi predstavljati prilično zamršen problem jer bi spajala kategorije karakteristične za deklinaciju (rod, broj, padež) i konjugaciju (lice) te bi se doista moglo dovesti u pitanje opravdanost temeljnih postavki ovoga opisa: klasifikacije prema tipovima promjene i jasnih kriterija te klasifikacije.

Na sreću sama GRAM nudi jednostavno rješenje kojim se zamjenička paradigma značajno pojednostavnjuje:

U ličnih je zamjenica kategorija lica izražena leksički (...) U glagola se kategorija lica morfologizirala, tj. ima svoj morfološki izraz u nastavku, iako sinkretički s kategorijom broja...<sup>37</sup>

U skladu s tom tvrdnjom različiti se oblici za pojedina zamjenička lica smješteni u različite leksikonske ulaze unutar kojih se može navesti podatak o licu potreban za obrade na višim jezičnim razinama. Istodobno je otpala potreba za otvaranjem još jedne vrste kompleksne paradigme za

---

<sup>36</sup> GRAM, str. 479.

<sup>37</sup> GRAM, str. 479.

koji je moglo biti upitno je li samo deklinacijska. Ovako je broj različitih kategorija u paradigmi sveden na tri: rod, broj, padež prema kojim je kategorijama promjena određena u deklinacijskim uzorcima.

#### 2.2.2.4. Komparativ i superlativ

Prihvativši rješenje o glagolskom vidu kao derivacijskoj kategoriji i promatrajući sve nastavke koji sudjeluju u hrvatskoj fleksiji sintetičkih oblika obično iskazivanih u tradicionalnim gramatikama može se primijetiti da se jedan (*naj-* u superlativu) od ostalih razlikuje dvama svojstvima:

- jedini se smješta ispred a ne kao svi ostali iza osnove
- nije fuzijski tj. ne objedinjuje više gramatičkih kategorija na planu sadržaja

Stoga naizgled neobično pitanje o uključivosti superlativnoga *naj-* u derivaciju odjednom prestaje takvim biti.<sup>38</sup> Dodatni argument u tome smjeru nudi GRAM:

Superlativ se **tvori** tako da se komparativu doda **prefiks** *naj-*...<sup>39</sup>

Dok bi se uporaba glagola *tvori* mogla prihvatiti u prenesenome značenju, imenica *prefiks* u ovoj definiciji nedvosmisleno upućuje na derivacijsko poimanje oblikovanja superlativnih oblika. Slične se formulacije javljaju i kod opisa komparativa:<sup>40</sup>

---

<sup>38</sup> Uzimajući u obzir ono što je rečeno gore uz bilješke 24 i 25.

<sup>39</sup> GRAM, str. 636. Isticanja moja.

<sup>40</sup> GRAM, str. 634.

»Nastavkom -ši *tvore* komparativ pridjevi...«, »Golema većina pridjeva *tvori* komparativ nastavkom -iji.«

Dodatni argument za smještanje komparativa i superlativa u domenu tvorbe riječi mogao bi biti i to što se na osnovu formiranu komparativnim i(li) superlativnim nastavcima dodaju nastavci deklinacijskoga uzorka koji se u tradicionalnoj gramatici zove deklinacija određenih pridjeva. Sasvim eksplicitno takvo tretiranje komparacije navodi i Babić (1986:12):<sup>41</sup>

Tako se npr. u gramatikama iz praktičnih razloga komparativ i superlativ (*kràsniji*, *nâjkrasniji*) smatraju oblicima riječi u pozitivu (*krâsan*), ali po tome što svaka ima svoju sklonidbu zapravo su nove riječi i mogle bi se ubrojiti u tvorbu, kao što se npr. ubraja pojačavanje svojstva sa *pre-* (*prèkrâsan*).

Ako su komparativ i superlativ nove riječi kojima se bavi derivacija, a ne fleksija, onda bi se morali nalaziti u leksikonu kao posebni novi ulazi. Međutim, argument da ti oblici imaju svoju sklonidbu ne čini se osobito jakim jer bi se u tom slučaju npr. u pridjeva određeni oblici mogli smatrati odjelitima od neodređenih oblika i tako uvrštavati kao posebni leksikonski ulazi. Što li tek reći za tako raznovrsne paradigme kao što su glagolske gdje bi se kudikamo lakše moglo naći slučajeva-kandidata za prelazak u derivaciju. Nadalje, dok se u navedenom citatu *pre-* doista može promatrati kao argument na strani derivacijskoga gledanja na superlativ, pitanje je može li se naći odgovarajuća podudarnost i za komparativ. Vjerojatnije je

---

<sup>41</sup> Aronoff (1976:92) komparaciju u engleskome također smješta u derivaciju. Zanimljiv problem u engleskome predstavlja opisna komparacija koji ni hrvatskome nije strana (GRAM, str. 637: *dobar*, *bolji/manje dobar/više dobar*, *najbolji/najmanje dobar/najviše dobar*), ali s obzirom na tako visok stupanj analitičnosti, primjereno je prepustiti je sintaksi, a ne morfologiji, pogotovo ne derivacijskoj. Tvrditi da su takve konstrukcije komparacija može se samo na osnovi značenja ukupne sintagme što je predmet sintakse *par excellence*.

da je razlika u sustavnosti, prije svega značenja, između *naj-* i *pre-* upravo onaj odlučujući čimbenik koji bi *pre-* ostavio u derivaciji, a *naj-* smjestio u fleksiju.<sup>42</sup>

Ako se ipak prihvati stajalište da su komparativ i superlativ dio fleksije, jedno od mogućih rješenja moglo bi biti uvođenje konfiksa za oblikovanje superlativa čime bi se uspio postići konzistentniji opis njegove fleksije. Međutim, ako se dopusti konfiks pri oblikovanju superlativa, zašto se ne bi dopustio pri oblikovanju svršenih glagola i zašto onda njih isključivati iz fleksije? Čini se da se odgovor nameće sam od sebe ukoliko se pokuša usporediti pravilnost i sustavnost ostvarivanja kategorije stupnjevanja. Komparativ se i superlativ ostvaruju samo jednim nastavkom,<sup>43</sup> odnosno, upravo nemogućnošću izbora između više nastavaka nasuprot mnogostrukim mogućnostima izbora u postupku perfektivizacije koje se mogućnosti ostvaruju prefiksacijom (svejedno je promatra li se kao samostalna ili konfiksno vezana) i(li) sufiksacijom. Nadalje, uz izbor prefiksa usko je vezana i razlika u značenju glagola koja ne pokazuje takve pravilnosti kao nastavak za komparativ i(li) superlativ i koja se razlika, bar što se tiče flektivne razine, može organizirati samo u obliku različitih leksikonskih ulaza,<sup>44</sup> za razliku od komparativa i(li) superlativa koji se u potpunosti mogu opisati relativno malim brojem uzoraka promjene.

Zbog jednostavnosti, a i gospodarnosti opisa, GENOBLIK se ne odriče tradicionalne opisne formulacije te paradigmu superlativa organizira upravo tako da svakome od oblika komparativne paradigme predmetne

---

<sup>42</sup> To tim prije jer se za pojačavanje svojstva pridjeva osim *pre-* može koristiti čitav niz drugih prefikasa: *arci-*, *arhi-*, *hiper-*, *super-*, *ultra-* i *vele-* navedenih u Babić (1986:445-446).

<sup>43</sup> v. pokušaj usustavljivanja u **3.2.2.3.**

<sup>44</sup> v. bilješku 29.



*naj-*.<sup>45</sup> Oblici se komparativne paradigme formiraju komparativnim nastavkom koji se potom uključuju u deklinacijski obrazac određenih pridjeva.

Sličnom se rješenju uteklo i pri opisu deklinacije neodređenih složenih zamjenica *tkogod*, *štogod*, *kakavgod*, *kojigod*, *čijigod* s time što se *-god* dodaje nakon obličnoga nastavka. Pitanje razdvojenoga pisanja oblika *kakavgod/kakav god*<sup>46</sup> moguće je riješiti fakultativnom transformacijom već spomenutom u bilješci 19.

Osobit slučaj predstavlja zamjenica *savkolik*, *svakolika*, *svekoliko* kod koje se sklanjaju obje sastavnice. S obzirom da bi to zahtijevalo značajno usložnjavanje deklinacijskih uzoraka, kao i dodatno kompliciranje računalne izvedbe modela, ona je ostavljena za neku bolju priliku.<sup>47</sup>

#### 2.2.2.5. Problem tvorbe priloga od pridjeva

Imajući u vidu ono što je dosad rečeno u točkama pod **2.2.2.** za odnos fleksije i derivacije problem bi morala predstavljati tvrdnja u Babić (1989) o tvorbi priloga od pridjeva bez obzira na to jesu li pridjevi u pozitivu, komparativu ili superlativu. Time se opis jedine promjene priloga — komparacije — rješava derivacijom. Nije dvojbeno da se prilozi moraju navesti u leksikonu kao zasebni leksikonski ulazi. No, ako se, unatoč argumentaciji iz **2.2.2.3.**, prihvati rješenje ponuđeno u Babić (1989) to s jedne strane vodi u nekonzistentnosti modela jer se tipovima promjene priključuje derivacija kao još jedan od tipova promjene oblika i time se

---

<sup>45</sup> O primjenljivosti tradicionalnoga pojma paradigme v. Carstairs (1988) i poglavlje **3**.

<sup>46</sup> GRAM, str. 659.

<sup>47</sup> Jedno od mogućih opravdanja za ovakav postupak mogla bi biti izoliranost slučaja, koji bi se mogao tretirati i kao sintaktička osobitost uz obaveznu transformaciju spajanja.

poništava granica između derivacije i fleksije. Istodobno se u leksikon moraju uključiti svi komparativni i superlativni oblici priloga kao zasebni leksikonski ulazi što nikako nije potvrđeno u dosadašnjoj praksi.<sup>48</sup> Ako se s druge strane prihvati rješenje prema kojem su prilozi već stvoreni od pridjeva i kao takvi imaju komparaciju, to vodi u dodatno usložnjavanje opisa i dvostruko navođenje pravila gdje se oblici srednjega roda pridjeva i oblici priloga podudaraju.

Ponovno je pragmatičnost nametnula odabir rješenja koje je usvojeno za GENOBLIK. Na teorijskoj se razini prilozi smatraju samostalnim leksikonskim ulazima s komparacijom, ali se isključivo je zbog efikasnije i brže obrade oblici različitih stupnjeva priloga uzimaju iz pridjevske paradigme kao gotovi oblici jednaki nominativu jednine srednjega roda pozitiva, komparativa ili superlativa pridjeva.

#### 2.2.2.6. Ostala ograničenja u fleksiji

S obzirom da su u suvremenome hrvatskom književnom jeziku oblici dvojine tj. maline u pismu isti ostalim oblicima paradigme u jednini i(li) množini, dvojinski dio paradigme nije uključen, uostalom, on se ne navodi ni u GRAM kada se iznose primjeri pojedinih uzoraka promjene.

Također se u ovome radu ne obrađuje osobit oblik imeničke fleksije, a to je fleksija kratica koja počesto sasvim izmiče pravilima. Ovdje se terminom *kratica* obilježuje ono što je u Babić-Finka-Moguš (1990:134) nazvano *kratica etiketa* tj. ono na što se u sklonidbi nastavak vezuje crticom. Naime, u tim je slučajevima rijetko kada potpuno jasno koja se od tih pokrata može, a koja ne može deklinirati. Ako se već može deklinirati, često je

---

<sup>48</sup> Osim za »nepravilne« komparative i superlative tj. komparative sa supletivnim osnovama: *dobro/bolje, zlo/gore, malo/manje, veliko/veće, dugo/dulje*.

nejasan i rod koji koja od kratica može imati itd. To područje niti jedna gramatika ne dodiruje premda bi, bar kao napomenu, valjalo navesti da se, npr. podatak o rodu i(li) deklinacijskom uzorku, može pronaći u leksikonu.<sup>49</sup>

### **2.2.3. Promjenljivost nasuprot nepromjenljivosti**

Već prije spomenuto da je opis načina na koji promjenljive riječi u hrvatskome mijenjaju svoje oblike cilj ovoga rada. Da bi se opisali različiti oblici promjenljivih riječi potrebno je, dakako, znati koje riječi mogu biti promjenljive, a koje ne. Već se pri tom elementarnom kriteriju može se vidjeti da u različitim autora postoje razlike. Dok se po pitanju imenica, pridjeva, zamjenica i glagola kao u potpunosti i nesumnjivo promjenljivih, i prijedloga, veznika, čestica i uzvika kao nesumnjivo nepromjenljivih, gramatike slažu,<sup>50</sup> prave kamene spoticanja u tom smislu predstavljaju prilozi i brojevi.

#### *2.2.3.1. Promjenljivost priloga*

Pitanje promjenljivosti priloga i različitih opisnih inačica vrlo je iscrpno obrađeno u Babić (1989).<sup>51</sup> Ovdje se, međutim, prilozi smatraju promjenljivim riječima s jednim tipom promjene — komparacijom. Argumentacija je za to zapravo ona ista koja se koristila pri smještanju

---

<sup>49</sup> Tadić (1992:10).

<sup>50</sup> Barić i drugi (1990:63), GRAM, str. 476-477.

<sup>51</sup> v. i u GRAM, str. 721.

komparacije pridjeva u sustav fleksije.<sup>52</sup> Ako se, dakle, komparacija promatra kao pojava u fleksiji, a ne derivaciji, onda se mora opisati i u pridjeva i u priloga. Time je osigurana neprotuslovnost i iscrpnost<sup>53</sup> opisa s obzirom na komparacijsku promjenu. Jednostavnost<sup>54</sup> bi pri vrednovanju različitih varijanata opisa morala biti posljednji od tri kriterija. Premda je, dakako, pitanje u kojem se dijelu jezičnoga opisa jednostavnost doista ostvaruje. Što je jednostavnije i gospodarnije: opisati komparaciju priloga uza svu njezinu složenost i uključivanje stanovitoga broja alternacijskih pravila (koja se jednostavno mogu »posuditi« od pridjeva) ili svaki od leksikonskih ulaza za oblik pozitiva priloga popratiti još s dva dodatna leksikonska ulaza koji uključuju komparativ i superlativ? S gledišta gramatičkoga opisa jednostavnije je i gospodarnije prikazati to predmetom tvorbe tj. smjestiti sve oblike u leksikon koji bi se time nužno morao usložniti jer bi takvi leksikonski ulazi morali upućivati jedni na druge. No opis jezika u svojoj ukupnosti, tek unutar koje valja procjenjivati gospodarnost opisa, ne uključuje samo propis nego i popis. Takav bi se popis, u slučaju uključivanja komparativa i superlativa u nj, znatno povećao jedinicama koje su lako obuhvatljive relativno jednostavnim proširenjem već postojećega propisa.

---

<sup>52</sup> v. 2.2.2.3.

<sup>53</sup> Hjelmslev (1980:20): »Opis ima biti neprotuslovan, iscrpan i što jednostavniji. Zahtjev za neprotuslovnošću nadređen je zahtjevu za iscrpnim opisom. Zahtjev za iscrpnim opisom nadređen je zahtjevu za jednostavošću.«

<sup>54</sup> Na nju se poziva Babić (1989:87): »Imamo kriterije za vrednovanje gramatičkih interpretacija. Bolje su one koje imaju manje jedinica, jednostavnija pravila, manje pravila koja obuhvaćaju veći broj primjera.«

### 2.2.3.2. Promjenljivost brojeva

Od svih vrsta riječi kako su definirane u tradicionalnih gramatikama, brojevi su najraznorodniji.<sup>55</sup> Prema istome semantičkom kriteriju u tu su vrstu skupljene riječi potpuno različita obličnoga »ponašanja«:

S kategorijalno-morfološkoga gledišta to je složena vrsta promjenjivih i nepromjenjivih riječi koje u jednom smislu pripadaju imenicama i pridjevima, a drugim su bliske priložima.<sup>56</sup>

Različitim uzorcima deklinacijske promjene mogu se opisati oni brojevi koji oblike mijenjaju poput imenica i(li) pridjeva, a bitno je istaknuti postojanje i beskonačnoga skupa sastavljenog od jednostavnih i složenih brojeva koje u većini odlikuje odsutnost promjene,<sup>57</sup> osim za složenice sa *jedan, dva, tri, četiri*. Isto navodi i GRAM:

Niz je prirodnih (cijelih) brojeva neograničen pa ih ima beskonačno mnogo i oni se izriču jednočlanim i višečlanim izrazima. Jednočlanih riječi nema mnogo, svega 48 (...) ***Kombinacijom tih riječi*** može se izraziti sve beskonačno mnoštvo brojeva.<sup>58</sup>

Dva su razloga zašto GENOBLIK ne obuhvaća složene brojeve.

Prvi: je li moguće zamisliti leksikon koji bi obuhvaćao beskonačan niz naziva za brojeve? Odgovor se nameće sam od sebe.

Drugi: ako se iz prethodnoga citata istaknuto primijeni dosljedno, onda složenim brojevima uopće nije mjesto u morfologiji već u sintaksi jer su

---

<sup>55</sup> Premda postoje i drukčija shvaćanja — npr. Pranjković (1992).

<sup>56</sup> GRAM, str. 663.

<sup>57</sup> v. Babić (1985) i Tafra (1989).

<sup>58</sup> GRAM, 664-665, isticanje moje.

kombinacije riječi sintagme koje u ovom slučaju imaju brojevno značenje.<sup>59</sup> Oni se i pravopisno tretiraju kao odvojene riječi,<sup>60</sup> a dodatni argument u tom smjeru predstavlja i mogućnost koordiniranja više sastavnica složenoga broja vezikom *i* za koju se mogućnost nikako ne bi moglo reći da pripada domeni morfologije. U tom je smislu na još jednom mjestu povučena granica između morfologije i sintakse.

### 2.2.3.3. Prijedlozi

Postoji još jedan slučaj kada neka riječ u hrvatskome posjeduju više oblika. Radi se o prijedlozima s naveskom: *beza, iza, ka, kroza, nada, niza, nuza, oda, poda, preda, sa, uza* koji postoje kao ekvivalenti *bez, iz, k, kroz, nad, niz, nuz, od, pod, pred, s, uz*. U GRAM stoje slijedeće tvrdnje:

Likovi s naveskom upotrebljavaju se zbog glasovnih ili morfoloških razloga. Zbog glasovnih se razloga upotrebljavaju da se izbjegne nemoguća ili nuobičajena glasovna skupina, gubljenje ili udvajanje suglasnika.<sup>61</sup>

Neki se prijedlozi upotrebljavaju i u okrnjenu liku, bez posljednjega samoglasnika: *napram < nasprama, naprot < naproti, pram < prama...*<sup>62</sup>

---

<sup>59</sup> Eric je Laporte na skupu *Computational Linguistics and Language Technology: Needs and Responses*, održanom 27-28. listopada 1989. u Dubrovniku pokazao da se brojevni izrazi u francuskome, kojih su složeni brojevi samo jedan dio, mogu obuhvatiti jedino potpunim sintaktičkim opisom.

<sup>60</sup> Osim u iznimnim slučajevima kao pri npr. novčanim transakcijama gdje je vjerojatno riječ o pravopisno reguliranoj zaštitnoj mjeri.

<sup>61</sup> GRAM, str. 726.

<sup>62</sup> GRAM, str. 727.

koje tvrdnje dopuštaju razliku u obliku nekih prijedloga. Ta se razlika, međutim, ne može smatrati dijelom fleksije prvenstveno zato što se takvi oblici u načelu nalaze u komplementarnoj distribuciji te se mogu smatrati alomorfima. Nadalje, ta razlika u obliku nipošto ne iskazuje ni leksičku ni sustavnogramatičku razliku u značenju.<sup>63</sup> Stoga ta pojava ne predstavlja zanimljivost za morfologiju.

#### 2.2.4. Sinteza nasuprot analizi

Premda postoje računalni sustavi koji se bave obradom morfologije prirodnih jezika u oba smjera — sintezom i analizom, barem su dva razloga zašto je GENOBLIK ograničen samo na sintezu.

Prvi je kompleksnost morfološke analize tj. »prepoznavanja« pripadnosti pojedinih oblika riječi iz korpusa njihovim paradigmama. Ta analiza nužno mora uključivati elemente sintakse<sup>64</sup> u slučajevima npr. složenih glagolskih vremena, neodređenih zamjenica razdvojenih enklitikama ili složenih brojeva tim prije što ti oblici često ne stoje u izravnom dodiru već su više ili manje udaljeni unutar iste ravnine.

Drugi je razlog taj što je GENOBLIK prvenstveno zamišljen za provjeru istraživačkih hipoteza, a njezin je smjer, barem za sada, odozgo prema dolje što je smjer sinteze tj. proizvođenja, a ne prepoznavanja oblika.

---

<sup>63</sup> Ovdje bi se moglo predložiti uvođenjem distinkcije između termina *oblik* kojim se obično obilježava jedna od stavki paradigme, a uključuje i plan izraza i plan sadržaja, te termina *lik* kojim bi se mogli obuhvatiti slučajevi poput ovoga kad promjena na planu izraza ne korespondira s promjenom na planu sadržaja. U tom slučaju promjena *lika* riječi nije morfološki uvjetovana.

<sup>64</sup> v. 2.2.1.

### 2.2.5. Realizacija postava pismom

S obzirom da je jedan od postavljenih zahtjeva bio realizacija postava standardnim zapisom tj. bez posebnih pismena, problem odnosa fonološkoga sustava nekoga jezika i njemu pridruženoga grafijskoga sustava i(li) pravopisnih pravila prisutan je i ovdje. On se pojavljuje na obje razine modela. I ulazna i izlazna razina modela u računalnoj izvedbi koriste se standardnim pravopisom uz manje iznimke koje su nužne iz razloga jednostavnije unutarnje organizacije obrade. Oblik zapisa ograničen je na standardni alfabet i ne dopušta konstrukte (osim za jat v. 2.2.5.2.) bez obzira na to imaju li npr. dijakronijskoga opravdanja, a koji bi se konstrukti kasnije morali uklanjati ili preoblikovati transformacijama.<sup>65</sup>

Nadalje, koliko god to ponekad bilo potrebno, u uporabi se standardnoga pravopisa ipak pretežito ne rabe naglasci stoga ih ni model GENOBLIK neće uzimati u obzir. Konačno, GENOBLIK ne daje klasifikaciju naglasnih tj. suprasegmentalnih nego segmentalnih uzoraka, a ni jedinice ni uzorci iz tih dviju klasifikacija često se ne poklapaju.

#### 2.2.5.1. Pisanje dž, lj, nj, št, žd

Prva iznimka od standardnoga pravopisa potreba je da se *dž, lj, nj* tretiraju kao jednoslovi tamo gdje je to nužno (npr. transformacija osnove *propnj-*

---

<sup>65</sup> v. Tadić (1992). Tipičan je takav konstrukt nepostojano -a- koje bi, da se zapisuje u osnovi, utjecalo na njezin izgled: *prašak < praš6k*, ali *praška < praš6ka*. Time se otvara potreba za transformacijom koja bi ukidala 6 u GDVLIjd i NDAVLI »kratke« množine i NGDAVLI »duge« množine. Rješenje, koje je ovdje prihvaćeno, komplementarno je, s istim rezultatom, ali uz isti broj preoblika treba kraći zapis osnove u leksikonu i manji broj operacija pri generiranju oblika paradigme — nepostojano se *a* ubacuje u Najd i Gmn.



umetanjem nepostojanoga -a- u *propanj*- mora *nj* tretirati kao jednoslov koji odgovara jednome fonemu kako operacija ne bi završila neovjerenom postavom \**propnaj*-). Međutim, u nekoliko slučajeva postojanje dvoslova upravo rezultira smanjenjem broja potrebnih transformacija (npr. Ijd imenice *jesen* koji glasi *jesenju* gdje se na izlaznoj razini dobiva grafijski ovjeren oblik jednostavnim dodavanjem nastavka -*ju* bez potrebe za izvođenjem transformacije koja bi se Akademijinom grafijom mogla zapisati ovako  $n + -ju > -nju$ ). Ovdje se, dakako, može uputiti prigovor za odstupanje od principa poštivanja jezičnih činjenica, no tih je odstupanja zanemarivo malen broj, a i sama su odstupanja isključivo izvedbene naravi, jer je zapis i u leksikonu i u konačnome rezultatu standardnopravopisan, a preznaka se obavlja samo za interne potrebe obrade.<sup>66</sup>

Drugi je slučaj, koji je mogao biti kandidat za odstupanje od standardnoga pravopisa, zapis skupova *št* i *žd*. U različitim oblicima promjene oni se, prema Babić (1986:209), vladaju poput nepčanih suglasnika. Silić (1992) pokazuje da su ti skupovi zapravo jesu nepčani suglasnici ako ih se promatra na drugoj razini tj. ne na fonološkoj nego na morfonološkoj. Pristup izložen u ovome radu mogao se prikloniti bilo kojem od rješenja jer postoje argumenti i za jedno i za drugo. Zapis *št* i *žd* kao dvoslova pojednostavnjuje unos podataka u leksikon jer se poklapa sa standardnom grafijom. S druge strane zapis *št* i *žd* kao jednoslova pojednostavnjuje klasifikaciju deklinacijskih uzoraka i uvjete okoline koji se moraju ispitati pri primjeni transformacija osnova. Zbog očekivanja da će podatke u leksikon unositi i lingvistički manje obrazovane osobe (studenti) — dakle iz praktičnih razloga — odabrano je prvo rješenje. No to nikako ne potire posebnost »ponašanja« skupova *št* i *žd* koje je ponašanje, uzeto kao jedan

---

<sup>66</sup> Takvo je rješenje primijenjeno u Tadić (1992), a sličnim se rješenjima utječu i drukčiji (dvorazinski) pristupi računalnoj obradi morfologije hrvatskoga kao npr. u Lopina (1992:189).

od kriterija, rezultiralo potrebom za dodatnim razgranjivanjem klasifikacije.

#### 2.2.5.2. *Pisanje jata*

Jedinu pravu iznimku od pravila standardnoga pravopisa čini upotreba grafema *ě* na ulaznoj razini za zapis jata koji se nalazi u korijenu osnove, a koji se jat na izlaznoj razini realizira različito na različitim mjestima u većem broju uzoraka promjene (npr. *vrěme* > *vrijeme*, *vremena*; *tělo* > *tijelo*, *tijela*, *tjelesa*; *viděti* > *vidjeti*, *vidim*).<sup>67</sup> Taj je problem riješen fakultativom transformacijom i posebnim zapisom jata u osnovi što je jedini slučaj odstupanja od standardnoga pravopisa u zapisivanju osnove u leksikonu.

#### 2.2.5.3. *Stanje pravopisa*

Konačno, ne bi bilo na odmet pripomenuti da je i sâm pravopis u ovome trenutku u stanju previranja. U Babić-Finka-Moguš (1994) dopušta se modifikacija oblika riječi u nekim elementima koji služe poglavito očuvanju ili povećanju zalihosti. Kod npr. imenica je to problem višestrukih uzastopnih transformacija osnove koje dovode do nerazabirljivosti što, dakako, zbog smanjene zalihosti, rezultira poteškoćama u komunikaciji. Ta je nerazabirljivost dvojaka:

1. *Homografna nerazabirljivost* gdje jedan te isti oblik može biti članom paradigmi leksikonski posve različitih riječi. Tipičan primjer jest oblik *paci* koji teoretski može biti Nmn od imenica

---

<sup>67</sup> Ovdje je preuzeto rješenje iz Tadić (1992).

*pac* (umak u kojem meso čeka na pripravu), \**patac*, *patak* (mužjak patke), *padac* (umanjenica od *pad*) i *padak* (umanjenica od *pad*).

2. *Nehomografna nerazabirljivost* gdje se kod rijetko rabljenih riječi često izgubi prepoznatljivost preobličene osnove (npr. *žuci* < *žutac*).<sup>68</sup>

Rješenje je čuvanje u pisanju posljednjega konsonanta osnove tamo gdje se to pokaže potrebnim.<sup>69</sup> Najčeći su slučajevi kod osnova na *-tac*, *-tak*, *-dac*, *-dak*: *pac/paci*, *patak/patci*, *padak/padci*. Pitanje je ima li smisla primijeniti to na riječi kojih paradigme ne sadrže nerazabirljive oblike ili na riječi koje su opće poznate i vrlo učestale tj. spadaju u jezgreni dio rječnika (npr. *otac/oca/\*otca*). Dakako da je takva vrsta nerazabirljivosti zapravo individualno određena i ovisi ponajprije o osviještenosti jezičnih pravila kao i o leksičkom fondu svakog pojedinog govornika.<sup>70</sup> No, GENOBLIK uključuje i takve slučajeve tj. dopušta oblike s neokrnljenom osnovom gdje bi se inače očekivala okrnjena (npr. *podatak/podatci*,<sup>71</sup> *paštitac/paštitca*<sup>72</sup>). Po kojem će se doista uzorku ravnati promjena oblika ovisi o tome koji se uzorak navede u leksikonu. Moguće nekom leksikonskom ulazu pridijeliti i više uzoraka i time dopustiti generiranje prema više različitih paradigmi polazeći od istoga leksikonskoga ulaza.

---

<sup>68</sup> Ovo je stav prof. dr. M. Moguša izložen u usmenoj komunikaciji.

<sup>69</sup> GRAMM str. 509, napomena na kraju t. 150. također, premda vrlo oprezno, dopušta i takvo rješenje.

<sup>70</sup> v. Tadić (1992).

<sup>71</sup> Potvrđen oblik za Nmn u okružnome dopisu Zajednice hrvatskih izdavača iz svibnja 1992. poslan na adresu Zavoda za lingvistiku Filozofskoga fakulteta u Zagrebu.

<sup>72</sup> Potvrđen oblik za Gjd u Šugar (1990), str. 89. natuknica *Cymóidum*, podnatuknica *I. monochasium*, potpodnatuknica *I. bostryx*.

### 3. TEORIJSKE POSTAVKE

Imajući u vidu cilj (2.1.) i ograničenja (2.2.) postavljena gore valjalo bi razmotriti teorijske postavke za stvaranje sustava GENOBLIK. One dolaze iz dvaju na prvi pogled nepomirljivih izvora: generativne i tradicionalne gramatike. Ta su dva izvora samo naizgled neudružljiva. Naime, Chomsky dopušta izravno uključivanje tradicionalnoga paradigmatskoga sustava promjene oblika, te navodi primjer postojanja uzorka deklinacije (tj. promjene: *declensional class* — DC) kao kategorije pridružene leksikonskom ulazu:

...we can restate the paradigmatic description directly in terms of syntactic features. Regarding each of the dimensions of the system of paradigms as a multivalued feature, with the specifications being not + and – but, let us say, integers conventionally associated with the traditional designators, (...) Notice that the specified features [1 Gender] and [1 DC] are *inherent* to this formative... (...) In short, the theory of syntactic features developed earlier can incorporate the traditional paradigmatic treatment directly.<sup>73</sup>

---

<sup>73</sup> Chomsky (1965:171-172): ...možemo iskazati paradigmatski opis izravno u obliku sintaktičkih obilježja. Promatrajući svaku od dimenzija sustava paradigmi kao viševrijednosno obilježje, s vrijednostima koje nisu + i – nego, recimo, cjelobrojnici uobičajeno pridruženi tradicionalnim gramatičkim oznakama, (...) Primjećuje se da su navedena obilježja [1 Rod] i [1 Deklinacijski uzorak] *inherentni* ovom formativu... (...) Ukratko, prije razvijena teorija sintaktičkih obilježja može izravno uključiti tradicionalnu paradigmatsku obradu. (prijevod moj).

Orijentiran na sintaktičku teoriju, on se i ne pokušava baviti tim problemom, ali ipak nudi preporuku:

Within our framework, either paradigmatic analysis in terms of features or sequential morphemic analysis is available, whichever permits the optimal and most general statement of some aspect of the syntactic or phonological system. It seems that in inflectional systems, the paradigmatic analysis has many advantages and is to be preferred...<sup>74</sup>

S obzirom na zahtjev za sposobnošću proizvodnja *svih mogućih* oblika riječi, prirodno je bilo teorijski okoliš za GENOBLIK potražiti u generativnoj gramatici. No imajući u vidu nezainteresiranost *standardne teorije* dane u Chomsky (1965) za pojave u morfologiji:<sup>75</sup>

Chomsky nije formulirao teoriju morfologije, već joj je samo priznao pravo postojanja utvrdivši da se struktura riječi razlikuje od strukture rečenice, da tvorba riječi počiva na različitim načelima od tvorbe rečenica i da stoga mora biti opisivana u drugom dijelu gramatike, a ne u sintaksi...<sup>76</sup>

---

<sup>74</sup> Chomsky (1965:174): Unutar našeg okvira istraživanja, moguća je ili paradigmatska analiza putem obilježja ili sekvencijalna morfemska analiza, kojagod dopušta optimalne i što općenitije tvrdnje o nekim vidovima sintaktičkoga i fonološkoga sustava. Čini se da u flektivnih sustava paradigmatska analiza ima mnoge prednosti i da bi je trebalo preferirati... (prijevod moj).

<sup>75</sup> v. također razmatranje o *leksikalističkoj hipotezi* Chomskoga u **2.2.1.** prema kojoj sintaksa nema pristupa na razine ispod razine riječi.

<sup>76</sup> Mihaljević (1991:82). *Tvorba* ovdje zacijelo stoji kao prijevod engleske riječi *formation* što dakako nije sinonim od *derivation*. Štoviše, Halle (1973) posvuda rabi termin *word-formation* koji uključuje i derivaciju i fleksiju. Ovdje bi bolji prijevod bio oblikovanje (formiranje) riječi i oblikovanje (formiranje) rečenica, a ne tvorba.

valjalo je naći teorijsku inačicu generativne gramatike koja se izričitije bavi morfologijom. Također je valjalo naći mjesto morfologiji unutar shematizirane organizacije generativne gramatike. U tom je smislu izuzetno važan rad Halle (1973) koji prvi donosi nacrt zasebne morfološke gramatičke sastavnice:

...the lexical insertion transformations have access only to words in the dictionary and, moreover, that the dictionary contains only (and all) fully inflected forms of the language. (...) One might propose that instead of inserting a single item, the lexical insertion transformations insert partial or entire paradigms, i.e. certain or all inflected forms of a given »word«. (...) ...there is every reason to expect that paradigms must appear as entities in their own right somewhere in a grammar. If my proposal is correct one such place would be the dictionary, from which the lexical insertion transformations draw items for insertion into a sentence. Note, incidentally, that if this is correct then the dictionary must be organized into paradigms in some way...<sup>77</sup>

Time je morfološka sastavnica smještena u leksikon,<sup>78</sup> ali je ujedno dan i prijedlog za njegovu organizaciju. Iz ovoga je citata moguće razabrati

---

<sup>77</sup> Halle (1973:9): ...transformacije leksičkoga umetanja imaju pristupa samo riječima u rječniku i, štoviše, rječnik sadrži samo (i sve) potpune flektivne oblike nekoga jezika. (...) Moglo bi se predložiti da umjesto umetanja jedne jedinice, transformacije leksičkoga umetanja umeću dijelove ili potpune paradigme tj. neke ili sve flektivne oblike dane »riječi«. (...) ...postoje svi razlozi za očekivanje da se paradigme kao samostalne jedinice moraju pojaviti negdje u gramatici. Ako je moj prijedlog točan, jedno od takvih mjesta bio bi rječnik, iz kojeg bi transformacije leksičkoga umetanja izvlačile jedinice za umetanje u rečenicu. Valja, međutim, primijetiti da, ako je to točno, tada rječnik mora biti na neki način paradigmatički organiziran... (prijevod moj).

<sup>78</sup> Valja napomenuti da Halle (1973) ne rabi termin *rječnik* umjesto u generativnih gramatičara uvriježenoga *leksikon*. On terminom leksikon označuje *sve moguće* riječi nekoga jezika tj. sve moguće kombinacije leksičkih, derivacijskih i flektivnih morfema, a terminom rječnik samo one oblike koji doista u jeziku postoje tj. koje su potvrđene.

važnost tradicionalnoga termina paradigme<sup>79</sup> koji postaje jedan od ključnih pojmova u flektivnoj morfologiji i njezinoj interakciji sa sintaksom upravo zahvaljujući mogućnosti njegove reinterpretacije u okviru sintaktičke teorije. Paradigmatski pristup nudi organizaciju leksikona prema kategorijama koje omogućuju odabir pojedinih oblika pri leksičkom umetanju u krajnje točke (»listove«) stablo-dijagrama koji predstavlja sintaktičku strukturu rečenica. To će umetanje uvijek biti umetanje jednoga oblika, ali hoće li se ono obaviti izravnim prenošenjem iz leksikona ili odabiranjem iz paradigme za ovaj rad i nije toliko važno jer je to pitanje odnosa morfologije i sintakse. Ono što je ovdje važno jest potreba za postojanjem paradigme kao teorijskoga konstrukta na nekoj jezičnoj razini.

Dakako, valja razlikovati dva sloja morfološke komponente:<sup>80</sup> derivacijski od flektivnoga. Paradoksalno je da ono što se u fleksiji obično smatra nepravilnošću, zapravo omogućuje njezino razlikovanje od derivacije:

One peculiarity of inflection is that it is *paradigmatic*. (...) Since derivational morphology is not paradigmatic it does not show any

---

<sup>79</sup> Trebalo bi ovdje točno odrediti što se u ovome radu razumije pod terminom *paradigma*. Za razliku od npr. Brozović-Ivić (1988:116) gdje su »paradigme nizovi sintetskih likova koji izražavaju određene kategorijske gramatičke odnose« i kojih u hrvatskom »ima osam« (paradigme padeža, gramatičkoga roda, pridjevske određenosti, gramatičkoga broja, gramatičkoga vremena, gramatičkoga lica, komparacijska paradigma te paradigma gramatičkoga načina), terminom *paradigma* u ovome radu obuhvaća se ukupnost sintetskih oblika u kojima se neka riječ ostvaruje. Paradigma se organizira prema osima koje predstavljaju gramatičke kategorije. Tako zamišljena paradigma može se modelirati višedimenzionalnom matricom koja svojim dimenzijama predstavlja gramatičke kategorije.

<sup>80</sup> v. 2.2.2. gore.

suppletion: that is, it does not concern itself with phonologically dissimilar but semantically related forms.<sup>81</sup>

Upravo zahvaljujući »snazi« paradigme tj. usustavljenoga polja kategorija u njoj mogu postojati i supletivni oblici. Paradigmatski je opis moguće primijeniti samo na fleksiju jer ona pokazuje znatno veću sustavnost od derivacije.

Imajući u vidu ono što je rečeno o razlici između derivacije i fleksije, moguće je zamisliti postojanje dviju razina obrade jezičnih jedinica u morfološkoj sastavnici gramatike.

Jednoj bi — derivacijskoj razini — ulaz činili popis leksičkih i derivacijskih morfema te pravila njihova kombiniranja, a izlaz bi bile *sve moguće* kombinacije leksičkih i derivacijskih morfema. Zbirni rezultat ili opis generiranja na razini te podsastavnice mogao bi se nazvati *ukupni derivacijski kapacitet* nekoga jezika.<sup>82</sup> Takva bi podsastavnica proizvodila nesamostalne jezične entitete na razini ispod razine riječi koji se u tradicionalnoj gramatici obično zovu osnovama.

Drugoj bi podsastavnici — flektivnoj — ulaz činili upravo popis osnova i gramatičkih morfema te pravila njihova kombiniranja. Izlaz bi te podsastavnice činile oblikovane riječi popraćene evaluiranim gramatičkim kategorijama koje kategorije predstavljaju pojedine dimenzije paradigme.

---

<sup>81</sup> Aronoff (1976:2): Jedna od neobičnosti fleksije je njezina *paradigmatičnost* (...) Budući da derivacijska morfologija nije paradigmatska ne pokazuje nikakve supletivnosti: to jest, ne bavi se fonološki različitim ali semantički povezanim oblicima. (prijevod moj).

<sup>82</sup> To bi u grubo odgovaralo onome što se u Halle (1973) nazivlje *leksikonom*.



Tako koncipirana paradigma zapravo je višedimenzionalna matrica<sup>83</sup> i u računalnome modelu ona je upravo tako i iskazana.<sup>84</sup>

Kako se ovaj rad bavi samo fleksijom, za njega je relevantna upravo ova druga i ovako koncipirana podsastavnica koja uključuje sve flektivne paradigme. Ako se prihvati prethodna preporuka Chomskoga, može se reći da je najsažetiji oblik opisa kombinacija osnova i nastavaka u flektivno bogatih jezika, kao što je hrvatski, zapravo *popis uzoraka promjene*. Stoga je *klasifikacija riječi prema načinima promjene* temelj lingvističkoga modela u ovome radu.<sup>85</sup>

### **3.1. Problemi klasifikacije vrsta riječi u hrvatskome**

Dva su temeljca svakoj klasifikaciji — predmet i kriteriji klasificiranja. Na ovome se mjestu preskače ponavljanje definicije riječi koja je navedena za potrebe ovoga rada<sup>86</sup> da bi se pažnja usmjerila na kriterije klasifikacije. Klasificiranje riječi postoji od prvih antičkih gramatika i sama se tradicionalna klasifikacija nije mnogo promjenila od njihova doba. Suvremeniji i formalniji pristupi jeziku nameću potrebe za klasifikacijama prema drukčijim kriterijima, ali:

---

<sup>83</sup> To koje su dimenzije (ili kategorije) paradigme relevantne za umetanje nekoga oblika riječi u neko mjesto u sintaktičkoj strukturi neke rečenice predstavlja gotovo nepresušno polje za daljnja istraživanja.

<sup>84</sup> v. o tome u Tadić (1992:66).

<sup>85</sup> v. 3.1. dolje.

<sup>86</sup> v. 2.2.1. gore.

Klasifikacija riječi je teoretsko lingvističko pitanje koje nije lako rješivo. To je jedna od onih »vječnih« lingvističkih tema kojoj se lingvisti stalno vraćaju. (...) Klasifikacije se i u drugim znanostima (sjetimo se samo zoologije i botanike, pa književnosti i dr.) stalno preispituju u traženju boljih. Jezične se klasifikacije moraju raditi u svakom jeziku posebno, na osnovi konkretnoga jezičnog materijala, jer ne postoji univerzalna klasifikacija za sve jezike. Jezici imaju nešto zajedničko, kao npr. imenice i glagole, ali i mnogo toga različitoga. Lingvisti ne mogu nikako pronaći općeprihvatljiv klasifikacijski kriterij za riječi.<sup>87</sup>

Takve su više ili manje slične tradicionalne klasifikacije rađene i za hrvatski u svakoj do sada objavljenoj slovnici ali:

(...) podjela na vrste riječi polidimenzionalnoga je karaktera i prilično elastična, a provodi se prema sintaktičkim, značenjskim i formalnim (morfološkim, katkad i tvorbenim) kriterijima, koji se u nekim konkretnim detaljima djelomično i prepleću. Imenice, pridjevi i prilozi dijele se na samostalne (prave), zamjeničke, brojevne i glagolske. Svaka od tih klasifikacijskih jedinica podliježe onda daljnjemu grananju, koje je već više u skladu s tradicionalnim diobama unutar pojedinih vrsta (...) podjela na vrste, u skladu s glavninom evropskoga gramatičarskoga naslijeđenog shvaćanja, zamjeničke riječi imeničkoga karaktera odvaja od imenica i zamjeničke riječi pridjevskoga karaktera od pridjevâ te ih oboje izdvaja kao posebnu vrstu riječi (zamjenice), s time da zamjeničke priloge ipak ostavlja među priložima, a isto tako brojevne imenice, pridjeve ali i priloge izdvaja također u posebnu vrstu (brojevi). (...) U svakom slučaju, svi ti ***tradicionalni postupci onemogućuju bilo***

---

<sup>87</sup> Tafra (1989:220).

***kakvu korektnu i dosljednu primjenu relevantnih  
klasifikacijskih kriterija za podjelu.***<sup>88</sup>

Temeljni je problem tradicionalne klasifikacije riječi na vrste čini se, zadržavanje naslijeđa grčkih, rimskih i srednjovjekovnih gramatičara. Očito je da zadržavanje tradicionalnih vrsta riječi, kao kategorija koje se mogu razlučiti jedino istodobnom primjenom (i to često nedosljednom) kriterija s više jezičnih razina, postaje veća smetnja no pomoć pri jezičnome opisu. Svijest o tome u našoj literaturi postoji već dovoljno dugo:

Ako je riječ o nekoj podjeli riječi, u prvom redu treba da nam bude jasno kakav je kriterij uzet za određenu podjelu: semantički, sintaktički, morfološki ili kakav drugi. U tradicionalnoj gramatici je vidljivo da se u istoj podjeli miješaju kriteriji. Npr. semantičke i sintaktičke kriterije susrećemo i u podjeli na vrste riječi i u podjeli na rečenične dijelove prema službi riječi u rečenici.<sup>89</sup>

Tako su za izlučivanje imenica, pridjeva i glagola bila bitna njihova morfološka obilježja, za zamjenice i brojeve semantička, a za prijedloge i veznike sintaktička. Zbog toga u toj klasifikaciji ima nekih nelogičnosti<sup>90</sup>

tako da je za hrvatski već bilo raznih pokušaja kritičkoga sagledavanja postojeće klasifikacije riječi, sa pozicije njihova promatranja kao leksičko-gramatičkoga razreda (Tafra 1989), sintaktičke samostalnosti (Pranjeković 1982), sintaktičke funkcionalnosti (Kordić 1992, Pranjeković 1992) ili diskurzivne funkcionalnosti (Velčić 1987). Stoga i ne čudi zahtjev:

---

<sup>88</sup> Brozović-Ivić (1988:115). Isticanja moja.

<sup>89</sup> Řehák (1968:10).

<sup>90</sup> Tafra (1989:220).

Trebalo bi za naš jezik, uzimajući u obzir što je lingvistika dosad postigla u svijetu i u nas, uraditi novu, dosljedniju klasifikaciju. Posebno je pitanje da li je moguća idealna klasifikacija bez ostatka. Vjerojatno nije moguća zato što će se uvijek, s koje god strane krenuli, naći neki »miješani« slučajevi, (...). Bez obzira na sve te teškoće, treba se prihvatiti posla.<sup>91</sup>

Općenito uzevši, da bi se taj posao mogao obaviti valja strogo razlikovati razine promatranja jer se time izbjegava opasnost od istodobne primjene kriterija kojima su ishodište različite jezične razine. A kad se razina promatranja odabere, treba paziti na dosljednu primjenu kriterija unutar iste razine.

Ovdje je odabrana razina na kojoj se riječ promatra kao kombinacija morfoloških objekata<sup>92</sup> ili morfoloških jedinica.<sup>93</sup> Ta je razina promatranja još uvijek ograničena na plan izraza, a na njoj se može objasniti zašto govornici hrvatskoga mogu relativno jednostavno razlučiti vrste riječi i dodijeliti im vrijednosti gramatičkih kategorija u slijedećem tekstu (bez obzira što im u tome djelomično može pomoći također, i upravo zato, raspoznatljiva sintaktička struktura):

*Bilo je kuhno i đipahne tovke  
na vabnjaku rovko zadronjaše;  
nemujne sasvim bjehu zorolovke  
i ručkale su šturnjače zdomašne!*<sup>94</sup>

---

<sup>91</sup> Tafra (1989:235).

<sup>92</sup> DiSciullo-Williams (1987), v. gore 2.2.1.

<sup>93</sup> Kačić (1992:122).

<sup>94</sup> Carroll (1985:136).

premda je sigurno da nitko ne razumije značenje onoga što je izrečeno.<sup>95</sup> U ovome je primjeru vidljivo kako je oblik u kojem se neka riječi javlja određen slijedom jedinica ispod razine riječi. Svakome, tko iole bolje poznaje hrvatski, jasno je da oblik *zadronjaše* predstavlja ili treće lice množine aorista ili drugo i(li) treće lice jednine imperfekta nekoga glagola premda je teško pretpostaviti da bi tko mogao prepoznati o kojem se točno glagolu radi. Ovdje se pokušava dati klasifikacija mogućih kombinacija klasa jedinica ispod razine riječi, koje kombinacije jedinica tvore ovjerene postavke. Te bi postavke govornici hrvatskoga mogli kao riječi uvrštati u sintaktičke strukture kao što je to vidljivo u gornjem slučaju koji je zanimljiv i slikovit upravo zbog svoje graničnosti.

Kada bi ishodišne epistemološke postavke ovoga rada bile slične onima u generativnoj gramatici, onda bi se mogla postaviti tvrdnja da ovaj rad pokušava dati model govornikova znanja toga jezičnog podskupa ili jezične razine. Međutim, GENOBLIK ne preuzimlje u potpunosti sve teorijske postavke generativne fonologije i(li) morfologije već samo one kojima se omogućuje sa što manje sredstava informatički učinkovito ostvarenje sustava za generiranje oblika riječi, a to su prije svega dvije razine (morfotaktička i preobličena) te postojanje transformacija (premda ne iste razine apstraktnosti i ista opsega). Ovaj je sustav daleko od svake pomisli da opisuje stvarno funkcioniranje ljudskoga jezičnog znanja — u toj su se metodološkoj zabludi premnogi već našli. Njegova je svrha u krajnjoj liniji sasvim praktične prirode — stvoriti sustav koji je kadar strojno proizvesti oblike riječi hrvatskoga standardnog jezika upotrebljive u sličnim sustavima za obradu ostalih jezičnih razina i(li) pojava.

---

<sup>95</sup> Zanimljivo je primijetiti kako su neke, u ovome slučaju evidentno funkcionalne, riječi kao što su pomoćni glagoli, veznici, prijedlozi i jedan kvantifikator (*sasvim*), morale izmaknuti poetskome postupku inače bi čitav iskaz bio potpuno neprepoznatljiv.

Problem zadržavanja tradicionalnogramatičkih vrsta riječi pri pokušaju računalne izvedbi takva jezičnoga opisa odvodi u računalno iznimno složene, ako ne i u gotovo neizvedive zahvate.

### **3.2. Dva postulirana konstrukta**

Pokazavši da je morfologiji mjesto u leksikonu, postavlja se pitanje kako najučinkovitije organizirati i prikazati sve podatke relevantne za izvođenje oblikâ neke riječi. Rješenje koje se, kad se problem opisa promjene oblikâ riječi u hrvatskome promatra izolirano od utjecaja s ostalih jezičnih razina, nadavalo samo od sebe, oblikovano je uvođenjem dvaju konstrukata. Na tim se konstruktima može temeljiti čitav opis te tako oni stječu status teorijskih postulata. Tako zamišljen opis promjene oblika riječi može se izvesti ako pretpostavimo postojanje:

1. *uzorka promjene*, konstrukta koji objedinjuje osobitosti u promjeni oblika nekoga skupa riječi;
2. *vrste promjene* koja je nadređena uzorku promjene. Po sastavu i kompleksnosti paradigmi podređenih uzoraka mogu se uočiti četiri temeljne vrste promjene oblika riječi u hrvatskome.<sup>96</sup> Svaka od temeljnih vrsta promjene:
  - obuhvaća više uzoraka promjene (npr. imenična deklinacija, ovisno o kriterijima klasifikacije, obuhvaća nekoliko stotina uzoraka promjene) ili

---

<sup>96</sup> v. detaljnije u 3.2.1.

- nekim uzorkom sudjeluje u formiranju složenih uzoraka promjene (složena glagolska vremena koja se sastoje od oblika konjugacije glavnoga glagola + oblika konjugacije pomoćnoga glagola; komparativ koji se sastoji od komparativnog nastavka + pridjevska deklinacija)

Uzorak promjene u GENOBLIK obuhvaća sve relevantne podatke o promjeni oblika riječi koje svoje flektivne oblike mijenjaju prema tome uzorku promjene. Svaki se uzorak promjene može zamisliti u obliku paradigmatske rešetke ili višedimenzionalne matrice kojom se predstavlja »svežanj« gramatičkih kategorija. Pojedinačni oblici riječi nalaze se na čvorovima rešetke. Uzorkom promjene određuju se tri komponente propisa nužne za generiranje nekoga oblika riječi:

1. odabir nastavaka: koji se nastavci moraju pridodati osnovi ne bi li se ostvario niz jedinica na morfotaktičkoj razini;
2. alternacijska pravila za preobliku osnove: koje se preoblike osnove moraju provesti ne bi li se morfotaktički niz realizirao na površinskoj razini
3. redoslijed primjene tih pravila.

Nastavci i pravila čine propis, a osnove smještene u leksikonu, popis. Takav je model promjene riječi moguće ostvariti računalnim sustavom.

U tu je svrhu bilo potrebno napraviti klasifikaciju uzoraka promjene koja je toliko fina da hvata sve nijansne razlike među njima. Takva klasifikacija mora biti osjetljiva na razne morfološke obavijesti<sup>97</sup> i one se tretiraju kao relevantni kriteriji klasifikacije uzoraka. U leksikonu je svakoj osnovi pri-

---

<sup>97</sup> cf. Mihaljević (1991:85), bilješka 122: »U morfološke obavijesti spadaju podaci kao što su: prisustvo ili odsustvo morfološke granice među sastavnicama, obilježja [strano], [domaće] i sl. i obilježja izuzetnosti.«.

družen broj uzorka promjene koji pripada jednoj od četiriju vrsta promjene. Time su svi podaci o fleksiji smješteni u leksikonski dio sustava GENOBLIK u najsžetijem obliku. Jedinu su dodatni podaci u leksikonu podatak o rodu imenica potreban tamo gdje nije predvidljiv iz klasifikacije, podatak koji ograničuje paradigmu imenica iznimkama tipa *plurale* i *singularum tantum*, a za glagole je to podatak o prijelaznosti tj. o mogućnosti da se za taj glagol izvede glagolski pridjev trpni.

Dopušten je i veći broj uzoraka za isti leksikonski ulaz tako da se za istu natuknicu oblici mogu mijenjati po:

- niti jednome uzorku promjene (*u* 0/0/0)
- jednom uzorku promjene (*bojazan* 392/0/0)
- više uzoraka promjene i to:
  - sudjelovanjem u više uzoraka iste vrste promjene (*iskati* 0/524,526/0 koji daje *ištem* i *iščem*)
  - sudjelovanjem u više uzoraka različitih vrsta promjene (*mastan* 456,457/0/441 koji daje *mastan*, *masni*, *masniji/najmasniji*)

### 3.2.1. Vrste promjene

Ovdje dana klasifikacija promjene oblika pretpostavlja postojanje četiri vrste promjene:

#### 3.2.1.1. Nulta promjena

Nultu promjenu karakterizira odsuće bilo kakve promjene oblika. Osnova je jednaka natuknici u leksikonskome ulazu. Mora se, međutim,



pretpostaviti zbog mogućnosti da se morfološka komponenta primijeni na one leksikonske ulaze koje u hrvatskome čine nepromjenljive riječi.

### 3.2.1.2. Deklinacija

Uzorci deklinacijske promjene obuhvaćaju promjene po:

1. vrsti -a
2. vrsti -e
3. vrsti -i
4. poimeničeni pridjevi
5. pridjevske promjene koje dalje obuhvaćaju
  - promjene neodređenoga oblika (pridjevi)
  - promjene određenoga oblika (pridjevi, zamjenice, brojevi)
  - zamjениčke promjene (zamjenice)
  - promjene brojeva

Izgled paradigme za uzorke vrste -a, -e, -i i uzorke poimeničenih pridjeva:

P1-P4	Jednina	Množina
Nominativ		
Genitiv		
Dativ		
Akuzativ		
Vokativ		
Lokativ		
Instrumental		

gdje P1-P4 znači da se navedena struktura ponavlja četiri puta zbog višestrukih likova za isti oblik (v. dolje **3.4.2.**). Kategorija roda pridružena je samome uzorku, a ukoliko iz njega nije izvediva, onda se mora posebno navesti u leksikonskome ulazu. Također se zbog njihove nepredvidljivosti iz samoga uzorka promjene u leksikonu mora navesti odsutnost jednine ili množine.

Izgled paradigme za uzorke promjene neodređenoga, određenoga oblika te zamjenica:

P1-P5	Muški rod		Ženski rod		Srednji rod	
Nominativ						
Genitiv						
Dativ						
Akuzativ						
Vokativ						
Lokativ						
Instrumental						

gdje P1-P5 znači da se navedena struktura ponavlja pet puta zbog istog razloga kao i paradigma imenične promjene.

Valja upozoriti na naizgled paradoksalnu situaciju da se promjena koja obuhvaća pridjeve i zamjenice naziva pridjevskom a ne, kako bi se u skladu s historijskom gramatikom očekivalo, zamjениčnom. Razlog je tome upravo težnja za razlikovanjem od tradicionalne podjele deklinacija na imeničnu i zamjениčnu. Razlog uključivanja promjene određenih oblika pridjeva, pridjevskih zamjenica i rednih brojeva u istu deklinaciju nametao se kao najjednostavnije rješenje koje istim sredstvima opisuje paradigme određene istim kategorijama — rod, broj, padež. Međutim, ovako postavljena pridjevska deklinacija uključuje i promjene koje pokazuju osobine tipične za tradicionalno određenu imeničnu deklinaciju kao što je promjena neodređenih oblika pridjeva (koji ionako pokazuju paradigmatiku izomofnost s određenim oblicima pridjeva) i promjena imeničnih zamjenica (*ja, ti, mi, vi, sebe...*). Razlog je jednostavan i praktičan. Pridjevska je deklinacija uključivši uzorke pridjevskih zamjenica morala uvesti čitav niz preoblika koje su karakteristične samo za zamjenice, koje preoblike najčešće pokrivaju supletivnost zamjениčkih oblika. No supletivnost je upravo u imeničnih zamjenica najizrazitija stoga bi njihovo uključivanje u uzorke imenične promjene rezultiralo znatnijim opterećenjem preobličnoga sustava koje se ondje rabi. Tako se u slučaju imeničnih zamjenica od ukupne paradigmatike matrice koristi samo onaj dio koji odgovara paradigmatiku imenične promjene.

Glavni su brojevi (*jedan, dva, tri, četiri*) kao i brojevne imenice (*oba, obadva, dvoje, troje, četvero/četvoro, petero/petoro...*) također uključene u pridjevsku deklinaciju s istim razlogom kao i imenične zamjenice.

### 3.2.1.3. Konjugacija

Konjugacijska paradigma, s obzirom da su u obzir uzeti samo jednostavni glagolski oblici<sup>98</sup> izgleda ovako:

	Prezent	Imperfekt	Aorist	Imperativ
1. lice jd.				
2. lice jd.				
3. lice jd.				
1. lice mn.				
2. lice mn.				
3. lice mn.				
	Pridjev radni	Pridjev trpni		
jd. m. r.				
jd. ž. r.				
jd. s. r.				
mn. m. r.				
mn. ž. r.				
mn. s. r.				
	Infinitiv	Prilog sadašnji	Prilog prošli	

Opreka je po vidu, u skladu s rečenim u **2.2.2.1.**, leksikalizirana tako da je podatak o ostvarljivosti imperfekta, aorista ili oba vezan uz uzorak.

Dvovidni se glagoli u tome slučaju obavezno uklapaju u dva ili više uzoraka, samo neki izuzetni slučajevi (*moći*) koji ionako čine zasebne uzorke omogućuju supostojanje oba vremena u istome uzorku. Podatak o glagolskoj prijelaznosti, ili točnije mogućnosti ostvaraja glagolskoga pridjeva trpnoga, semantičke je prirode tako da se podatak o tome mora navesti u leksikonu.

Da su u obradu na istoj razini bili uključeni složeni glagolski oblici glagolska bi paradigma bila znatno složenija, štoviše, to bi dovelo do

---

<sup>98</sup> v. gore **2.2.2.2.**

višestrukoga povećanja broja oblika jer bi za svaki složeni oblik trebalo uključiti sve promjene oblika sastavnica. Samo bi perfekt (glagolski pridjev radni + nenaglašeni oblici prezenta *biti*) imao: 3 roda x 3 lica x 2 broja = 18 oblika. GRAM (str. 715) dopušta i uporabu naglašanih oblika u oblikovanju perfekta te bi stoga valjalo napraviti mjesta za novih 18 oblika. Kakvoj to kombinatornoj eksploziji vodi vidljivo je odmah na prvi pogled. To je bio još jedan od razloga za prepuštanje opisa složenih glagolskih vremena nekom drugom radu.

#### 3.2.1.4. Komparacija

Paradigma komparacije sastoji se od dvaju oblika koji zatim sami postaju osnova koja zatim:

1. izvodi oblike prema jednom od uzoraka određene pridjevske promjene ili
2. izvodi oblike stupnjeva priloga:

Pozitiv			
Komparativ			
Superlativ			
\$462	Muški rod		Ženski rod
Nominativ			
Genitiv			
Dativ			
Akuzativ			
Vokativ			
Lokativ			
Instrumental			

gdje \$462 označuje broj uzorka određene pridjevske promjene, a Pozitiv, Komparativ i Superlativ oblike priloga.

### 3.2.2. Uzorci promjene

Vrsta promjene ostvaruju se kroz uzorke promjene. Ti uzorci nasljeđuju osobine svoje vrste u smislu njima obuhvaćene paradigme.

#### 3.2.2.1. Deklinacija

Za deklinaciju kriteriji su razvrstavanja uzoraka promjene preuzeti iz tradicionalne klasifikacije uzoraka kakva je dana u GRAM str. 489-667. Osobita je pozornost poklonjena kriterijima koji se eksplicitno navode prije popisa deklinacijskih uzoraka, a potom se u samome popisu uzoraka ne pokazuje njihova primjena. Kriteriji klasifikacije uzoraka promjene imenica i njihova primjena potanko su objašnjeni u Tadić (1992:44-50) stoga bi se ovdje valjalo zadržati samo na osnovnom.

Redoslijed primjene kriterija za imeničnu deklinaciju u GENOBLIK je manje-više isti kao i u GRAM s tom razlikom što je primjena strukturno-gramatičkih kriterija živo/neživo i mn1/mn2 (množina/množina s umetkom) obvezatna, štoviše, kad se u klasifikaciji iscrpu kriteriji sastava osnove i(li) kriteriji »domaće«/»strano«, kao obvezatan dolazi grozd uzoraka koji ima opći oblik:

razina x. sastav osnove i(li) »domaće«/»strano« kriterij

razina x+1. živo

razina x+2. mn1 \$n

razina x+2. mn12 \$n

razina x+1. neživo

razina x+2. mn1 \$n

razina x+2. mn12 \$n

gdje je  $n$  broj uzorka. Ta je obvezatnost rezultat zahtjeva za pokrivanjem svih oblika imenica *in potentia* makar se odmah i ne pronašle potvrde za neke uzorke.

Kriteriji klasifikacije pridjevske deklinacije također slijede one predložene u GRAM s tom razlikom da je zbog kompleksnosti, ali i izomorfnosti, paradigma neodređenih oblika pridjeva odvojena od paradigme određenih oblika. Pridjevima koji mogu imati i određene i neodređene oblike stoga se u leksikonu pridružuju dva uzorka promjene, po jedan za svaku vrstu oblikâ. Time je ujedno i riješen problem odsutnosti neodređenih oblika jer je takvim pridjevima pridružen samo jedan broj uzorka.

### 3.2.2.2. Konjugacija

Sustav je konjugacijskih uzoraka promjene klasificiran prema kriterijima drukčijim od onih u tradicionalnoga razvrstavanja na šest (GRAM) i(li) sedam (Barić i dr. (1990)) vrsta sa razredima u pojedinim vrstama. Temeljna je razdioba na plodne i neplodne glagole prema kriterijima:

Plodni su glagoli izrazno definirani kao glagoli koji u neprefigiranom infinitivu imaju najmanje tri sloga, za razliku od neplodnih glagola koji imaju dvosložan neprefigiran infinitiv (tzv. I. vrsta, korijenska).<sup>99</sup>

gdje se neplodni u većini slučajeva poklapaju sa I. vrstom kako je ona određena u GRAM (str. 681-688). Za opis je plodnih glagola iz Babić (1991) preuzeta koncepcija temeljne strukture glagolske osnove koja može završiti četirima samoglasima: *a*, *i*, *ě*, *e*. Takva struktura osnove tj. njezin dočetni samoglasnik određuje konačni izgled i »ponašanje« kako same osnove tako i nastavaka. U strogom generativnofonološkom pristupu koji se favorizira

---

<sup>99</sup> Babić (1991:9).

u Babić (1991) težište je opisa prebačeno na pomno određivanje fonoloških pravila za alternacije, kako osnova tako i nastavaka, te određivanje njihova stroga redoslijeda. Opis koji je ovdje primijenjen, temeljen na klasifikaciji uzoraka promjene, čini se jednostavniji. Za plodne glagole postavlja četiri glavna uzorka prema dočetnim samoglasima, a konačnih je uzoraka nakon uključivanja obličnih inačica (imerfekt na *-ah* i(li) *-ijah*, glagolski pridjev radni na *-nuo* i(li) *-(a)o*, imperativ na *-j0* ili *-(j)i*) ukupno 43.

Pri izvođenju oblika plodnih glagola rabilo se svega 15 krajnje jednostavnih alternacijskih pravila koja u najvećem broju slučajeva djeluju potpuno neovisno o okolini u kojoj se moraju primijeniti. Osim toga, alternacije se izvode u malom broju uzastopnih alternacijskih koraka koji su potrebni za generiranje nekoga oblika: najviše ih je potrebno četiri, a i to svega u dva slučaja. S neplodnim je glagolima situacija složenija te je i raznolikost upotrijebljenih preoblika veća (21) premda ne znatno.

Tako koncipirana klasifikacija omogućila je još jedan oblik pojednostavnjenja opisa glagolske paradigme a to je izbjegavanje višestrukih nizova nastavaka za izvođenje oblika po istoj vremenskoj ili načinskoj paradigmi. Tako npr. za prezent GRAM navodi četiri niza nastavaka kojima se izvode oblici:

jd.	1. l.	-ēm	-jēm	-īm	-ām
	2. l.	-ēš	-jēš	-īš	-āš
	3. l.	-ē	-jē	-ī	-ā
mn.	1. l.	-ēmo	-jēmo	-īmo	-āmo
	2. l.	-ēte	-jēte	-īte	-āte
	3. l.	-ū	-jū	-ē	-āju <sup>100</sup>

dok su GENOBLIK oni svi svedeni na jedinstven niz:

---

<sup>100</sup> GRAM, str. 680.

-m, -š, -0, -mo, -te, -u

od kojega odstupanja postoje u doista malom broju slučajeva posebnih uzoraka za glagole *hotjeti/hoću* i *moći/mogu*.<sup>101</sup> Takvu je unifikaciju bilo moguće provesti zbog prihvaćene pretpostavke o postojanju samoglasa na kraju osnove (čak i u neplodnih glagola!) koji je samoglas ovisno o uzorku podložen preoblikama. Nije, dakle, potrebno voditi računa o tome koje vrste i koji glagolski razredi izvode prezent kojim od četiriju nizova nastavaka. Istovrsna su objedinjavanja provedena i u drugim dijelovima paradigme:

	Prezent	Imperfekt	Aorist	Imperativ
1. lice jd.	-m	-ah	-h	
2. lice jd.	-š	-aše	-0	-0
3. lice jd.	-0	-aše	-0	
1. lice mn.	-mo	-asmo	-smo	-mo
2. lice mn.	-te	-aste	-ste	-te
3. lice mn.	-u	-ahu	-še	
	Pridjev radni	Pridjev trpni		
jd. m. r.	-o	-n, -t		
jd. ž. r.	-la	-na, -ta		
jd. s. r.	-lo	-no, -to		
mn. m. r.	-li	-ni, -ti		
mn. ž. r.	-le	-ne, -te		
mn. s. r.	-la	-na, -ta		
	Infinitiv	Prilog sadašnji	Prilog prošli	
	-ti, -ći	-ći	-vši	
			-v	

Za razliku od Babić (1991:52-54) nije se ulazilo u daljnju analizu obličnih nastavaka (*-a-* u imperfektu kao oznake trajnosti, *-h* u aoristu i imperfektu kao oznake prošlosti...) jer bi to moralo uključiti primjenu preoblika i na nastavcima, a GENOBLIK to ne dopušta jer je primjena preoblika ograničena samo na osnove.

<sup>101</sup> U uzorcima 503-514, 517-523 *-u* u 3. l. mn. zamijenjeno je s *-0* iz razloga sasvim praktične prirode — smanjivanje broja preoblika.



### 3.2.2.3. Komparacija

Ako isključimo nekoliko pridjeva sa supletivnim osnovama (*dobar/bolji, zao/gori, malen/manji, velik/veći, dug/dulji/duži*), opći je oblik nastavka za komparativ na leksičkoj razini:

osnova + X-j + nastavak

gdje X može biti (-s-, -l-, -o-, -i-), a u nastavak se mogu uključiti nastavci uzorka deklinacije određenih pridjeva ili nastavak za prilog. Umetak za komparativ -j- djeluje na prethodne suglasnike jotacijom. S obzirom da -j- uvijek djeluje preoblično, te je njegovo navođenje u nastavku u uzorku promjene uz istodobno navođenje preobliske koja rezultira njegovim postojanjem, zapravo zalihosno, iskorišten je drukčiji oblik zapisa imajući u vidu ograničenja nametnuta pravopisnim pravilima.<sup>102</sup> To je u računalnome modelu zbog praktičnih razloga rezultiralo različitim nastavcima iako oni u lingvističkome opisu pokazuju jednoobraznost (+X-j-). Osim jotacije na djelu su u komparativa i druge preobliske osnove: obezvučavanje, bipalatalizacija, alternacija *l/o*, kraćenje skupova *stn* i *ždn*, a opća je i pravilno provedena osobina svih komparativnih oblika kraćenje jata u korijenu. Oblici superlativa jednaki su komparativnim oblicima s predmetkom (ili dijelom konfiksa za superlativ) *naj-*.

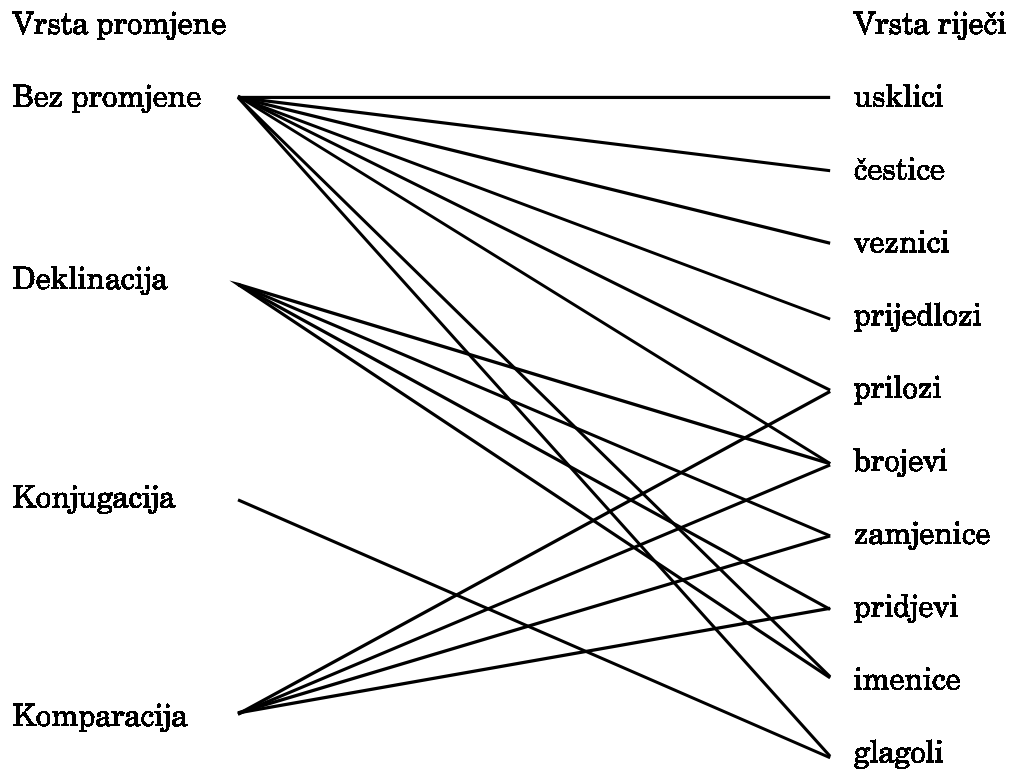
### 3.3. Odnosi tipova promjene prema tradicionalnim vrstama riječi

Slijedećim su grafikonom prikazani odnosi između tradicionalnih vrsta riječi i u 3.2. postuliranih vrsta promjene. Valja upozoriti da, premda glagolski pridjevi imaju likove koji odgovaraju likovima oblika nominativa

---

<sup>102</sup> v. 2.2.5.1. gore.

jednine i množine svih triju rodova, oni se ipak ne smatraju dijelom pridjevske paradigme. Naime, padež kao paradigmatska kategorija koja je najrelevantnija za deklinaciju uopće, u konjugaciji ne postoji, stoga se ne uspostavlja odnos između deklinacije i glagola. To što se mnogi glagolski pridjevi, koji se uostalom i zovu pridjevima, odvajaju od glagolske paradigme i postoje kao samostalni pridjevi i nije činjenica od presudnoga značaja za uspostavljanje te relacije. Štoviše, diskusija o prirodi konjugacije kao jedinstvene paradigme<sup>103</sup> mogla bi obuhvatiti mnoge radove i veće i značajnije od ovoga.




---

<sup>103</sup> v. 2.2.2.2. gore i bilješku 34.

### 3.4. Razine obrade

Promatrano sa strane realizacije računalnoga modela, svi se do sada ostvareni sustavi razlikuju prema:<sup>104</sup>

1. područjima morfologije koja su kadri obraditi (neki se bave samo fleksijom, neki derivacijom, a neki obuhvaćaju oba područja)
2. osnovnim teorijskim (pret)postavkama (ovisno o tome s koje se strane pristupa problemu — s lingvističke ili informatičke)
3. načinima računalne izvedbe (dva su osnovna načina na koja se dosada pokušavalo rješavati problem: algoritamski i relacijski)<sup>105</sup>
4. »smjerom« obrade (neki se bave samo analizom, neki samo sintezom, a neki pokrivaju oba »smjera«)

Valjalo bi prvo naglasiti da svi dosadašnji sustavi ravnaju jedinicama isključivo na planu izraza. Dakle, kad se gdje govori o morfemima, misli se zapravo na njihove morfove.

#### 3.4.1. Dvije razine obrade

Točka 3. gore zaslužuje nešto više pažnje od ostalih točaka. Riječ je rezultat obrade u svim sustavima bez obzira na put kojim se do nje došlo, a polazna je pretpostavka također svih sustava postojanje jedinica nižih od razine riječi koje se kombiniraju na neki način ne bi li oformile bazičnu strukturu te riječi. Nazovimo tu razinu *morfotaktičkom razinom*.<sup>106</sup> Neki sustavi već

---

<sup>104</sup> cf. Tadić (1992:18).

<sup>105</sup> Stohastički će se pristupi rješavanju problema (v. u Kržak 1990) izbjeći jer bi lingvista prvenstveno trebale zanimati lingvističke (kojima algoritmi i relacije ne bi smjeli biti strane), a ne isključivo statističke metode rješavanja problema.

<sup>106</sup> cf. Jäppinen-Ylilampi (1986), str. 258.

u ishodištu jedinice razine ispod razine riječi ne izjednačuju s jezičnim jedinicama. Takvi su pretežito sustavi koji kreću s informatičkim teorijskim postavkama, a one nerijetko rezultiraju više ili manje čistim modelom relacijske datobaze. Takav je sustav, koji ne vodi računa o segmentaciji riječi na morfeme već se služi postupkom *krnjenja* (truncating) i potom kombiniranjem tako dobivenoga početka s više dočetaka, opisan u Kržak (1988). U čistim relacijskim modelima morfologije postoje samo sastavnice na nižoj razini i njihovi odnosi. Time otpada potreba za algoritmima kojima se opisuju preoblike nekih od jedinica ispod razine riječi. Takvi, međutim, sustavi zapravo ne mogu opisati promjene tipa *vuk/vuče/vuci* a da se ili ne ogriješe o jezično ispravnu segmentaciju riječi na sastavnice ili ne potroše ogromne količine memorije za pohranjivanje svih transformiranih oblika morfema tj. alomorfa kao zasebnih početaka.<sup>107</sup> Zbog toga takvi sustavi, a njihovi su tvorci najčešće informatičari, za svoju znanost potpuno legitimnim postupkom, a za lingvistiku kršenjem nekih od osnovnih lingvističkih postulata, pokušavaju uštedjeti na prostoru i vremenu obrade tako da oblike iste paradigme reduciraju na zajednički nepromjenjivi početak (npr. *vu-*) i više promjenljivih dočetaka (*-k*, *-ci*, *-kova*, *-če* itd.) čime se doista dobiva na učinkovitosti obrade. Tu, međutim, dolazi do lingvistički apsurdnih situacija gdje je osnova *0-* a nastavak cijela riječ *-slati* ili *-šaljem* jer nema niti jednoga zajedničkog pismena s početka tih dvaju oblika.

Nasuprot relacijskim modelima, algoritamski modeli, pored osnovne morfotaktičke relacije kojom se uspostavlja temeljni niz jedinica ispod razine riječi, uključuju i primjenu algoritama kojima se neke od jedinica u tom nizu preobličuju. Time se pokazuje potreba za lučenjem dviju razina obrade i dviju razina podataka:

---

<sup>107</sup> cf. Tadić (1992:19).

1. *morfotaktička razina* na kojoj su jedinice ispod razine riječi tek povezane relacijom u *morfotaktički niz*. Ova razina služi kao ulazna za preoblike.
2. *preobličena razina* na kojoj su neke jedinice u nizu transformirane i koji niz predstavlja gotovu *riječ*. Riječ je rezultat djelovanja transformacija na morfotaktički niz ili, preciznije, na jedan dio morfotaktičkoga niza.<sup>108</sup>

Od jedinica koje formiraju niz na prvoj razini postoji barem jedna koja je element zatvorenoga skupa i barem jedna koja je element otvorenoga skupa. Zbog te pripadnosti barem jedne jedinice otvorenome skupu ni za jedan se sustav ne može tvrditi da opisuje oblike svih riječi nekoga jezika već samo one oblike čije su sastavnice uvrštene u popis elemenata toga potencijalno otvorenoga skupa, a koji je popis *via facti* uvijek konačan. No zato se može pokušati opisati sve tipove generiranja oblika riječi, a njih je zacijelo konačan broj. Teško je zamisliti da bi se u npr. derivaciji hrvatskih riječi moglo beskonačno rekurzivno nizati prefikse ili sufikse. Tà jedan je jednostavan razlog tome: riječi da bi bile funkcionalne moraju biti upotrebljive tj. ostvarljive u konačnom vremenu.

U GENOBLIK-u sastavnice koje sudjeluju u formiranju morfotaktičkoga niza nisu iste naravi. Jedinica koja je element zatvorenoga skupa pripada popisu gramatičkih morfema (morfova gramatičkih morfema) tj. u tradicionalnoj terminologiji flektivnih nastavaka dok jedinica koja je element otvorenoga skupa zapravo i nije leksički morfem (morf leksičkoga morfema) nego ono što se u tradicionalnoj terminologiji označuje kao osnova koja može biti jednaka ili morfu leksičkoga morfema ili bilo kojoj dopuštenoj kombinaciji leksičkoga i derivacijskoga (ili derivacijskih)

---

<sup>108</sup> v. bilješku 11 gore.

morfema. U GENOBLIK djelovanje preoblika ograničeno je isključivo na osnovu jer je to razina od koje sustav počinje s obradom.

### 3.4.2. Postave na dvije razine

Na morfotaktičkoj se razini flektivni oblici deklinacije i jednostavni oblici konjugacije mogu prikazati općom operacijom nizanja sa + kao operatorom spajanja i := kao operatorom pridruživanja:<sup>109</sup>

morfotaktički niz := osnova + nastavak

gdje se osnova uzimlje iz leksikona, a nastavak iz uzorka promjene koji pripada toj osnovi što je u konkretnom slučaju izvedeno u obliku popisa razmještaja nastavaka za sve oblike svih deklinacijskih i konjugacijskih uzoraka. Na preobličenoj je razini to pridruživanje već obavljeno:

riječ := preobličena osnova + nastavak

gdje je osnova preobličena u skladu s preoblikama specifičnim za taj oblik u paradigmi toga uzorka, a koje se preoblike uzimlju iz popisa svih preoblika za sve oblike svih uzoraka promjene. Te su dvije razine u ovom računalnom rješenju povezane samo u jednom smjeru: od morfotaktičke k preobličenoj razini.<sup>110</sup>

Za razliku od jednostavne operacije nizanja kakvom se izvode oblici prve i druge vrste promjene, uzorci komparacije pokazuju složenije oblike nizanja sastavnica za komparativ.

morfotaktički niz := (osnova + nastavak<sub>1</sub>) + nastavak<sub>2</sub>

---

<sup>109</sup> cf. Tadić (1992:39).

<sup>110</sup> Pokušaj rješavanja toga odnosa u drugome smjeru – što bi omogućilo analizu oblika – mogao bi biti ostvarivim jednostavnom inverzijom popisa nastavaka koji bi se »odbijali« od riječi, a potom bi trebalo pronaći osnove koje odgovaraju preoblikovanom ili nepreoblikovanom ostatku riječi.

ili za superlativ:

$$\text{morfotaktički niz} := (\text{nastavak}_{1a} + \text{osnova} + \text{nastavak}_{1b}) + \text{nastavak}_2$$

gdje  $\text{nastavak}_1$  pripada komparacijskome uzorku, a  $\text{nastavak}_2$  deklinacijskome. U slučaju složenih glagolskih oblika operacija pridruživanja izgleda ovako:

$$\text{morfotaktički niz} := (\text{osnova}_1 + \text{nastavak}_1) \& (\text{osnova}_2 + \text{nastavak}_2)$$

ili:

$$\text{morfotaktički niz} := ((\text{osnova}_1 + \text{nastavak}_1) \& (\text{osnova}_2 + \text{nastavak}_2)) \& (\text{osnova}_3 + \text{nastavak}_3)$$

gdje jedinice sa indeksom jedan ili jedan i dva pripadaju pomoćnom glagolu ili povratnoj zamjenici. Znakom & obilježen je operator pridruživanja bez spajanja.

Kad je riječ o osnovi, jedan od temeljnih zahtjeva koji je bio postavljen na početku, bila je upotreba samo jedne osnove za jedan uzorak promjene, ali i samo jedne osnove za jednu natuknicu. Drugim riječima, za sve oblike koji čine izlaz iz nekoga uzorka promjene, ulaz, osim u uzorku raspoređenih nastavaka, čini samo jedna jedinstvena osnova kojoj se dodjeljuju nastavci. Za imenice, pridjeve i plodne glagole<sup>111</sup> to i nije *novum*, ali, koliko se to moglo razabrati iz postojeće literature, neplodni glagoli još nisu bili promatrani na ovaj način.<sup>112</sup>

---

<sup>111</sup> Premda tradicionalni gramatički opis još uvijek rabi dvije osnove — infinitivnu i prezentsku — Babić (1991) je pokazala da je za plodne glagole u hrvatskome moguće postulirati postojanje jedinstvene ishodišne osnove.

<sup>112</sup> Jedine su iznimke od ovoga pravila, dakako, pomoćni glagoli *biti* i *htjeti* zbog svoje doista visoko izražene supletivnosti.

Kad je riječ o nastavcima valja imati na umu da jednostavno pridruživanje osnove i temeljnih nastavačnih likova ne rješava probleme izvođenja oblika. Naime, postavlja se pitanje kako tretirati umetke kao što su npr. množinski umetak *-ov-/-ev-* kod imenica.<sup>113</sup> Je li to zasebna morfološka jedinica ili dio nastavka? Ako nije zasebna morfološka jedinica, kako to da se pojavljuje razlika u značenjima između oblika s umetkom i bez njega (*akti* 'spisi'/*aktovi* 'slike nagoga ljudskog tijela', *puci* 'svjetine'/*pukovi* 'pukovnije'<sup>114</sup>)? Ako je pak umetak zasebna morfološka jedinica, zašto je onda intuitivno gledano jasno da takva množina pokazuje visok stupanj pravilnosti, štoviše, bez umetka, nastavci su isti, kao i značenje u najvećem broju slučajeva. Pri realizaciji modela pojavilo se pitanje odluke kako će se tretirati umetke, kao dio transformacija ili nastavaka? Zbog uštede u broju transformacija u svakom padežu u deklinaciji, bilo je potrebno umetke tretirati kao dio nastavka tj. uvesti »složene« nastavke. Time su se osim umetka za obilježavanje množine mogle riješiti i neke drugi oblici »umetanja« u deklinaciji (dometanje *-god* u uzorcima kojima se opisuje deklinacija neodređenih zamjenica) Dakle, struktura nastavka može se prikazati kao:

nastavak := <umetak> + jednostavni nastavak

ili

nastavak := jednostavni nastavak + <dometak>

gdje je unutar < i > element kojeg ovisno o padežu i uzorku ima ili nema.<sup>115</sup>

U konjugaciji je bilo moguće malobrojne umetke (umetanje *-v-* kod glagolskih pridjeva trpnih: *izuti/izut*, *izuven* i umetanje epentetskoga *-l-*

---

<sup>113</sup> cf. Tadić (1992:40).

<sup>114</sup> cf. Samardžija (1988:132).

<sup>115</sup> Za popis imeničnih nastavaka v. Tadić (1992:41).



također u glagolskih pridjeva trpnih: *iscrpiti/isrpljen*), tretirati kao preoblike što je, kao što se i željelo, rezultiralo manjim popisom nastavaka.

Daljnjem je usložnjavanju problema svakako pridonijela činjenica da za pojedina mjesta u paradigmatskoj matrici tj. za pojedine oblike u hrvatskome postoji više mogućih likova (u deklinaciji: Ijd i-vrste, Gmn u više uzoraka a- i e-vrste, cijeli dijelovi paradigme u dvojnim oblicima: množina s umetkom ili bez njega za istu imenicu itd.; dvojni oblici za imperfekt ili glagolski pridjev trpni u pojedinim glagolskim razredima). Najveća moguća uočena raznolikost u imeničnoj deklinaciji potvrđena je u Gmn imenice *zub* (*zubi, zubiju, zuba, zubova*) i Nmn imenice *zec* (*zeci, zečevi, zecevi, zecovi*). U pridjevskoj deklinaciji, koja uključuje i zamjenice, najveću raznolikost pokazuje Djd srednjega roda zamjenice *moj, moja, moje* (*mojem, mojemu, momu, mome, mom*) i Djd muškoga i srednjega roda zamjenice *koji, koja, koje* (*kojem, kojemu, komu, kome, kom*).

Stoga se za deklinaciju primijenilo rješenje koje uključuje postojanje četiriju odnosno pet *paralelnih paradigmi* od kojih je samo prva obavezno ispunjena (čak ni ona potpuno u slučajevima *plurale* i *singulare tantum*), a ostale tri ili četiri po potrebi.

U konjugaciji je zbog većega broja oblika u polaznoj paradigmi u odnosu na deklinaciju (37 kod dvovidnih glagola nasuprot 14 kod imenica) odabrano rješenje prema kojem se glagol, ako ima dvojne ili višestruke likove za jedan oblik, mijenja po dvama ili više uzorcima promjene s time da se isti oblici naknadno brišu. Jednako je rješenje primijenjeno i u slučaju komparacije tako da komparativi s više likova za isti oblik mijenjaju po više komparacijskih uzoraka.

## 4. KLASIFIKACIJA UZORAKA PROMJENE

U ovom se poglavlju rada daje pregled klasifikacije korištene u sustavu GENOBLIK.

Upotrijebljene kratice: C = suglasnik, C1 = prvi suglasnik, C2 = drugi suglasnik gledano slijeva, V = samoglasnik, <> = različito, m = muški rod, f = ženski rod, n = srednji rod, inf. = infinitiv, prez. = prezent, impf. = imperfekt, aor. = aorist, impr. = imperativ, prad. = pridjev radni, ptrp = pridjev trpni, prpr. = prilog prošli, prsa = prilog sadašnji, komp. = komparativ, sup. = superlativ. Popis pismena ili skupina pismena uvijek je u [ ]. Navode se i brojevi uzoraka (\$ ispred broja) i brojevi uzoraka iz GRAM (# ispred broja) kako bi se mogle uočiti razlike u klasifikacijama. Riječi koje ispred sebe imaju pisme # služe kao primjeri za uzorak.

### 4.1. *Nulta promjena*

Nulta promjena obilježena je uzorkom \$0. Leksikonski ulazi kojima je pridružen uzorak \$0 ne podliježu promjenama i njihova se obrada zaobilazi.

## 4.2. Deklinacija

### 4.2.1. a-vrsta

#### 1. a-vrsta

##### 1.1. imenice m

##### 1.1.1. imenice m s -0

1.1.1.1. dočetak na C (prednepčani-c) tj. [p, b, v, f, m, n, t, d, s, z, l, r] (osim *car* i *put*, višesložnih imenica na -âr, -îr, -êr, -er, -êz, -îz, -ûz, te imenica na -št, -žd)

1.1.1.1.1. dočetak na -VC (V <> nepostojano a) ili -VCC (CC = [st, zd])

1.1.1.1.1.1. živo (posebno *mrav*, *crv*, *gost*, *čovjek*)

1.1.1.1.1.1.1. mn1 \$1 #1

1.1.1.1.1.1.2. mn12 \$2 #2

1.1.1.1.1.1.3. osnova u jd na -in- \$3 #3

1.1.1.1.1.1.4. *Turčin* \$4

1.1.1.1.1.1.5. *crv*, *mrav* \$5 #4

1.1.1.1.1.1.6. *gost* \$6 #5

1.1.1.1.1.1.7. *čovjek* \$7 #6

1.1.1.1.1.2. neživo (osim *zub*, *prst*, *par*, *sat*, *hvat*, *ar*)

1.1.1.1.1.2.1. mn1 \$8 #7

1.1.1.1.1.2.2. mn12 \$9 #8

1.1.1.1.1.3. *zub* \$10 #9

1.1.1.1.1.4. *prst* \$11 #10

1.1.1.1.1.5. *par*, *sat*, *hvat* \$12 #11

1.1.1.1.1.6. *ar* \$13 #12

1.1.1.1.2. dočetak na -CaC (a = nepostojano)

1.1.1.1.2.1. osnova na -CaC (a = nepostojano), C1 & C2 iste zvučnosti ili C1 i/ili C2 = sonant (-lad, -laf, -čam, -dam, -jam, -lam, -ram, -tam, -zam, -dan, -van, -bar, -dar, -far, -gar, -har, -kar, -par, -tar, -var, -vas, -cat, -hat, -kat, -mat, -pat, -rat) i *pas* i *san*

1.1.1.1.2.1.1. živo

1.1.1.1.2.1.1.1. mn1 \$14 #13

1.1.1.1.2.1.1.2. mn12 \$15 #14

1.1.1.1.2.1.2. neživo (posebno *nokat*)

1.1.1.1.2.1.2.1. mn1 \$16 #15

1.1.1.1.2.1.2.2. mn12 \$17 #16

1.1.1.1.2.1.2.3. *nokat* \$18 #17

1.1.1.1.2.2. osnova na -CaC (a = nepostojano), C1 & C2 različite zvučnosti (obezvučena osnova u kosim padežima)

1.1.1.1.2.2.1. živo

1.1.1.1.2.2.1.1. mn1 \$19

1.1.1.1.2.2.1.2. mn12 \$20

1.1.1.1.2.2.2. neživo

1.1.1.1.2.2.2.1. mn1 \$21

1.1.1.1.2.2.2.2. mn12 \$22 #18

1.1.1.1.3. dočetak na -CC (CC <> [st, zd])

1.1.1.1.3.1. osnova na -kt, -nt, -pt, -rt

1.1.1.1.3.1.1. bez nepostojanoga a u Gmn

- 1.1.1.1.3.1.1.1. živo
  - 1.1.1.1.3.1.1.1.1. mn1 \$23
  - 1.1.1.1.3.1.1.1.2. mn12 \$24
- 1.1.1.1.3.1.1.2. neživo
  - 1.1.1.1.3.1.1.2.1. mn1 \$25
  - 1.1.1.1.3.1.1.2.2. mn12 \$26
- 1.1.1.1.3.1.2. sa i bez nepostojanoga a u Gmn
  - 1.1.1.1.3.1.2.1. živo
    - 1.1.1.1.3.1.2.1.1. mn1 \$27 #19
    - 1.1.1.1.3.1.2.1.2. mn12 \$28
  - 1.1.1.1.3.1.2.2. neživo
    - 1.1.1.1.3.1.2.2.1. mn1 \$29 #20
    - 1.1.1.1.3.1.2.2.2. mn12 \$30 #akt #pakt #sekt
- 1.1.1.1.3.2. osnova na ostale nenepčane skupove
  - 1.1.1.1.3.2.1. bez nepostojanoga a u Gmn
    - 1.1.1.1.3.2.1.1. živo
      - 1.1.1.1.3.2.1.1.1. mn1 \$31
      - 1.1.1.1.3.2.1.1.2. mn12 \$32 #hund
    - 1.1.1.1.3.2.1.2. neživo
      - 1.1.1.1.3.2.1.2.1. mn1 \$33
      - 1.1.1.1.3.2.1.2.2. mn12 \$34 #šund
    - 1.1.1.1.3.2.1.3. gumb \$35
  - 1.1.1.1.3.2.2. sa i bez nepostojanoga a u Gmn
    - 1.1.1.1.3.2.2.1. živo
      - 1.1.1.1.3.2.2.1.1. mn1 \$36 #21
      - 1.1.1.1.3.2.2.1.2. mn12 \$37
    - 1.1.1.1.3.2.2.2. neživo
      - 1.1.1.1.3.2.2.2.1. mn1 \$38 #22
      - 1.1.1.1.3.2.2.2.2. mn12 \$39
- 1.1.1.2. dočetak na C(mekonepčani)
  - 1.1.1.2.1. dočetak na -k
    - 1.1.1.2.1.1. osnova na -Vk (V <> nepostojano a)
      - 1.1.1.2.1.1.1. »domaće« s alomorfnom osnovom u Nmn
        - 1.1.1.2.1.1.1.1. živo
          - 1.1.1.2.1.1.1.1.1. mn1 \$40 #23
          - 1.1.1.2.1.1.1.1.2. mn12 \$41 #24
        - 1.1.1.2.1.1.1.2. neživo
          - 1.1.1.2.1.1.1.2.1. mn1 \$42 #25
          - 1.1.1.2.1.1.1.2.2. mn12 \$43 #26
      - 1.1.1.2.1.1.2. posuđenice bez alomorfne osnove u Nmn
        - 1.1.1.2.1.1.2.1. živo
          - 1.1.1.2.1.1.2.1.1. mn1 \$44
          - 1.1.1.2.1.1.2.1.2. mn12 \$45
        - 1.1.1.2.1.1.2.2. neživo
          - 1.1.1.2.1.1.2.2.1. mn1 \$46 #lek
          - 1.1.1.2.1.1.2.2.2. mn12 \$47
    - 1.1.1.2.1.2. osnova na -ak (a = nepostojano)
      - 1.1.1.2.1.2.1. osnova na -Cak (a = nepostojano, C = [p, f, š, j, lj, m, n\*, nj, r\*, v] \* (osim *sinak* i *zajutrak*))
        - 1.1.1.2.1.2.1.1. živo
          - 1.1.1.2.1.2.1.1.1. mn1 \$48 #27
          - 1.1.1.2.1.2.1.1.2. mn12 \$49 #28
        - 1.1.1.2.1.2.1.2. neživo

1.1.1.2.1.2.1.2.1. mn1 \$50 #29  
 1.1.1.2.1.2.1.2.2. mn12 \$51 #30  
 1.1.1.2.1.2.2. *sinak* i *zajutrak* (a = nepostojano)  
 1.1.1.2.1.2.2.1. *sinak* \$52 #31  
 1.1.1.2.1.2.2.2. *zajutrak* \$53  
 1.1.1.2.1.2.3. osnova na -lak (a = nepostojano)  
 1.1.1.2.1.2.3.1. -l- ostaje  
 1.1.1.2.1.2.3.1.1. živo  
 1.1.1.2.1.2.3.1.1.1. mn1 \$54 #32  
 1.1.1.2.1.2.3.1.1.2. mn12 \$55  
 1.1.1.2.1.2.3.1.2. neživo  
 1.1.1.2.1.2.3.1.2.1. mn1 \$56  
 1.1.1.2.1.2.3.1.2.2. mn12 \$57 #33  
 1.1.1.2.1.2.3.2. -l- > -o-  
 1.1.1.2.1.2.3.2.1. živo  
 1.1.1.2.1.2.3.2.1.1. mn1 \$58 #volak (volci, voci)  
 1.1.1.2.1.2.3.2.1.2. mn12 \$59  
 1.1.1.2.1.2.3.2.2. neživo  
 1.1.1.2.1.2.3.2.2.1. mn1 \$60 #34  
 1.1.1.2.1.2.3.2.2.2. mn12 \$61  
 1.1.1.2.1.2.4. osnova na -cak (a = nepostojano)  
 1.1.1.2.1.2.4.1. živo  
 1.1.1.2.1.2.4.1.1. mn1 \$62 #35  
 1.1.1.2.1.2.4.1.2. mn12 \$63  
 1.1.1.2.1.2.4.2. neživo  
 1.1.1.2.1.2.4.2.1. mn1 \$64 #36  
 1.1.1.2.1.2.4.2.2. mn12 \$65  
 1.1.1.2.1.2.5. osnova na -čak i -ćak (a = nepostojano)  
 1.1.1.2.1.2.5.1. živo  
 1.1.1.2.1.2.5.1.1. mn1 \$66 #37  
 1.1.1.2.1.2.5.1.2. mn12 \$67 #38  
 1.1.1.2.1.2.5.2. neživo  
 1.1.1.2.1.2.5.2.1. mn1 \$68 #39  
 1.1.1.2.1.2.5.2.2. mn12 \$69 #40  
 1.1.1.2.1.2.6. osnova na -sak (a = nepostojano)  
 1.1.1.2.1.2.6.1. živo  
 1.1.1.2.1.2.6.1.1. mn1 \$70  
 1.1.1.2.1.2.6.1.2. mn12 \$71 #41  
 1.1.1.2.1.2.6.2. neživo  
 1.1.1.2.1.2.6.2.1. mn1 \$72 #42  
 1.1.1.2.1.2.6.2.2. mn12 \$73 #43  
 1.1.1.2.1.2.7. osnova na -tak (a = nepostojano)  
 1.1.1.2.1.2.7.1. okrnjena osnova u Nmn  
 1.1.1.2.1.2.7.1.1. živo  
 1.1.1.2.1.2.7.1.1.1. mn1 \$74 #44  
 1.1.1.2.1.2.7.1.1.2. mn12 \$75 #45  
 1.1.1.2.1.2.7.1.2. neživo  
 1.1.1.2.1.2.7.1.2.1. mn1 \$76 #46  
 1.1.1.2.1.2.7.1.2.2. mn12 \$77 #47  
 1.1.1.2.1.2.7.2. neokrnjena osnova u Nmn  
 1.1.1.2.1.2.7.2.1. živo  
 1.1.1.2.1.2.7.2.1.1. mn1 \$78  
 1.1.1.2.1.2.7.2.1.2. mn12 \$79  
 1.1.1.2.1.2.7.2.2. neživo

1.1.1.2.1.2.7.2.2.1. mn1 \$80  
 1.1.1.2.1.2.7.2.2.2. mn12 \$81  
 1.1.1.2.1.2.8. osnova na -stak, -štak (a = nepostojano)  
 1.1.1.2.1.2.8.1. živo  
 1.1.1.2.1.2.8.1.1. mn1 \$82  
 1.1.1.2.1.2.8.1.2. mn12 \$83  
 1.1.1.2.1.2.8.2. neživo  
 1.1.1.2.1.2.8.2.1. mn1 \$84 #48  
 1.1.1.2.1.2.8.2.2. mn12 \$85  
 1.1.1.2.1.2.9. osnova na -Cak (C = zvučni tj. [b, d, dž, đ, g, z, ž])  
 1.1.1.2.1.2.9.1. osnova na -Cak (C = [b, ž])  
 1.1.1.2.1.2.9.1.1. živo  
 1.1.1.2.1.2.9.1.1.1. mn1 \$86 #49  
 1.1.1.2.1.2.9.1.1.2. mn12 \$87  
 1.1.1.2.1.2.9.1.2. neživo  
 1.1.1.2.1.2.9.1.2.1. mn1 \$88 #50  
 1.1.1.2.1.2.9.1.2.2. mn12 \$89  
 1.1.1.2.1.2.9.2. osnova na -dak  
 1.1.1.2.1.2.9.2.1. okrnjena osnova u Nmn  
 1.1.1.2.1.2.9.2.1.1. živo  
 1.1.1.2.1.2.9.2.1.1.1. mn1 \$90 #51  
 1.1.1.2.1.2.9.2.1.1.2. mn12 \$91  
 1.1.1.2.1.2.9.2.1.2. neživo  
 1.1.1.2.1.2.9.2.1.2.1. mn1 \$92 #52  
 1.1.1.2.1.2.9.2.1.2.2. mn12 \$93  
 1.1.1.2.1.2.9.2.2. neokrnjena osnova u Nmn  
 1.1.1.2.1.2.9.2.2.1. živo  
 1.1.1.2.1.2.9.2.2.1.1. mn1 \$94  
 1.1.1.2.1.2.9.2.2.1.2. mn12 \$95  
 1.1.1.2.1.2.9.2.2.2. neživo  
 1.1.1.2.1.2.9.2.2.2.1. mn1 \$96  
 1.1.1.2.1.2.9.2.2.2.2. mn12 \$97  
 1.1.1.2.1.2.9.3. osnova na -zdak, -ždak (a = nepostojano)  
 1.1.1.2.1.2.9.3.1. živo  
 1.1.1.2.1.2.9.3.1.1. mn1 \$98  
 1.1.1.2.1.2.9.3.1.2. mn12 \$99  
 1.1.1.2.1.2.9.3.2. neživo  
 1.1.1.2.1.2.9.3.2.1. mn1 \$100 #53  
 1.1.1.2.1.2.9.3.2.2. mn12 \$101  
 1.1.1.2.1.2.9.4. osnova na -đak  
 1.1.1.2.1.2.9.4.1. živo  
 1.1.1.2.1.2.9.4.1.1. mn1 \$102  
 1.1.1.2.1.2.9.4.1.2. mn12 \$103  
 1.1.1.2.1.2.9.4.2. neživo  
 1.1.1.2.1.2.9.4.2.1. mn1 \$104 #54  
 1.1.1.2.1.2.9.4.2.2. mn12 \$105  
 1.1.1.2.1.2.9.5. osnova na -zak osim *brizak, drozak, mozak*  
 1.1.1.2.1.2.9.5.1. živo  
 1.1.1.2.1.2.9.5.1.1. mn1 \$106 #55  
 1.1.1.2.1.2.9.5.1.2. mn12 \$107  
 1.1.1.2.1.2.9.5.2. neživo

1.1.1.2.1.2.9.5.2.1. mn1 \$108 #56  
 1.1.1.2.1.2.9.5.2.2. mn12 \$109  
 1.1.1.2.1.2.9.6. imenice *drozak* i *brizak*, *mozak*  
 1.1.1.2.1.2.9.6.1. *drozak* \$110 #57  
 1.1.1.2.1.2.9.6.2. *mozak*, *brizak* \$111 #58  
 1.1.1.2.1.3. osnova na -Ck  
 1.1.1.2.1.3.1. »domaće« s alomorfnom osnovom u Nmn  
 1.1.1.2.1.3.1.1. živo  
 1.1.1.2.1.3.1.1.1. mn1 \$112  
 1.1.1.2.1.3.1.1.2. mn12 \$113  
 1.1.1.2.1.3.1.2. neživo  
 1.1.1.2.1.3.1.2.1. mn1 \$114 #59  
 1.1.1.2.1.3.1.2.2. mn12 \$115 #60  
 1.1.1.2.1.3.2. posuđenice bez alomorfne osnove u Nmn  
 1.1.1.2.1.3.2.1. živo  
 1.1.1.2.1.3.2.1.1. mn1 \$116 #Bask  
 1.1.1.2.1.3.2.1.2. mn12 \$117  
 1.1.1.2.1.3.2.2. neživo  
 1.1.1.2.1.3.2.2.1. mn1 \$118  
 1.1.1.2.1.3.2.2.2. mn12 \$119  
 1.1.1.2.2. dočetak na -g  
 1.1.1.2.2.1. osnova na -Vg (V <> nepostojano a)  
 1.1.1.2.2.1.1. »domaće« s alomorfnom osnovom u Nmn  
 1.1.1.2.2.1.1.1. živo  
 1.1.1.2.2.1.1.1.1. mn1 \$120 #61  
 1.1.1.2.2.1.1.1.2. mn12 \$121 #62  
 1.1.1.2.2.1.1.2. neživo  
 1.1.1.2.2.1.1.2.1. mn1 \$122 #63  
 1.1.1.2.2.1.1.2.2. mn12 \$123 #64  
 1.1.1.2.2.1.2. posuđenice bez alomorfne osnove u Nmn  
 1.1.1.2.2.1.2.1. živo  
 1.1.1.2.2.1.2.1.1. mn1 \$124 #mag  
 1.1.1.2.2.1.2.1.2. mn12 \$125  
 1.1.1.2.2.1.2.2. neživo  
 1.1.1.2.2.1.2.2.1. mn1 \$126  
 1.1.1.2.2.1.2.2.2. mn12 \$127  
 1.1.1.2.2.2. osnova na -ag (a = nepostojano)  
 1.1.1.2.2.2.1. živo  
 1.1.1.2.2.2.1.1. mn1 \$128  
 1.1.1.2.2.2.1.2. mn12 \$129 #65  
 1.1.1.2.2.2.2. neživo  
 1.1.1.2.2.2.2.1. mn1 \$130  
 1.1.1.2.2.2.2.2. mn12 \$131 #66  
 1.1.1.2.2.3. osnova na -Cg  
 1.1.1.2.2.3.1. »domaće« s alomorfnom osnovom u Nmn  
 1.1.1.2.2.3.1.1. živo  
 1.1.1.2.2.3.1.1.1. mn1 \$132 #67  
 1.1.1.2.2.3.1.1.2. mn12 \$133  
 1.1.1.2.2.3.1.2. neživo  
 1.1.1.2.2.3.1.2.1. mn1 \$134 #68  
 1.1.1.2.2.3.1.2.2. mn12 \$135 #69  
 1.1.1.2.2.3.2. posuđenice bez alomorfne osnove u Nmn  
 1.1.1.2.2.3.2.1. živo  
 1.1.1.2.2.3.2.1.1. mn1 \$136

- 1.1.1.2.2.3.2.1.2. mn12 \$137
- 1.1.1.2.2.3.2.2. neživo
  - 1.1.1.2.2.3.2.2.1. mn1 \$138 #erg
  - 1.1.1.2.2.3.2.2.2. mn12 \$139
- 1.1.1.2.3. dočetak na -h
  - 1.1.1.2.3.1. osnova na -Vh (V <> nepostojano a)
    - 1.1.1.2.3.1.1. »domaće« s alomorfnom osnovom u Nmn
      - 1.1.1.2.3.1.1.1. živo
        - 1.1.1.2.3.1.1.1.1. mn1 \$140 #70
        - 1.1.1.2.3.1.1.1.2. mn12 \$141 #71
      - 1.1.1.2.3.1.1.2. neživo
        - 1.1.1.2.3.1.1.2.1. mn1 \$142 #72
        - 1.1.1.2.3.1.1.2.2. mn12 \$143 #73
    - 1.1.1.2.3.1.2. posuđenice bez alomorfne osnove u Nmn
      - 1.1.1.2.3.1.2.1. živo
        - 1.1.1.2.3.1.2.1.1. mn1 \$144
        - 1.1.1.2.3.1.2.1.2. mn12 \$145
      - 1.1.1.2.3.1.2.2. neživo
        - 1.1.1.2.3.1.2.2.1. mn1 \$146
        - 1.1.1.2.3.1.2.2.2. mn12 \$147
  - 1.1.1.2.3.2. osnova na -ah (a = nepostojano)
    - 1.1.1.2.3.2.1. živo
      - 1.1.1.2.3.2.1.1. mn1 \$148 #74
      - 1.1.1.2.3.2.1.2. mn12 \$149
    - 1.1.1.2.3.2.2. neživo
      - 1.1.1.2.3.2.2.1. mn1 \$150
      - 1.1.1.2.3.2.2.2. mn12 \$151
  - 1.1.1.2.3.3. osnova na -Ch
    - 1.1.1.2.3.3.1. »domaće« s alomorfnom osnovom u Nmn
      - 1.1.1.2.3.3.1.1. živo
        - 1.1.1.2.3.3.1.1.1. mn1 \$152 #75
        - 1.1.1.2.3.3.1.1.2. mn12 \$153 #76
      - 1.1.1.2.3.3.1.2. neživo
        - 1.1.1.2.3.3.1.2.1. mn1 \$154
        - 1.1.1.2.3.3.1.2.2. mn12 \$155
    - 1.1.1.2.3.3.2. posuđenice bez alomorfne osnove u Nmn
      - 1.1.1.2.3.3.2.1. živo
        - 1.1.1.2.3.3.2.1.1. mn1 \$156
        - 1.1.1.2.3.3.2.1.2. mn12 \$157
      - 1.1.1.2.3.3.2.2. neživo
        - 1.1.1.2.3.3.2.2.1. mn1 \$158 #bronh
        - 1.1.1.2.3.3.2.2.2. mn12 \$159
- 1.1.1.3. dočetak na C(nepčani) tj. [ć, đ, š, ž, č, dž, j, lj, nj] te -št i -žd
  - 1.1.1.3.1. dočetak na -CC ili -VC (V <> nepostojano a) osim jedno- i dvosložnih osnova na -eC
    - 1.1.1.3.1.1. dočetak na -CC (CC <> [ št, žd] ) ili -VC
      - 1.1.1.3.1.1.1. živo
        - 1.1.1.3.1.1.1.1. mn1 \$160 #77
        - 1.1.1.3.1.1.1.2. mn12 \$161 #78
      - 1.1.1.3.1.1.2. neživo
        - 1.1.1.3.1.1.2.1. mn1 \$162 #79
        - 1.1.1.3.1.1.2.2. mn12 \$163
    - 1.1.1.3.1.2. dočetak na -CC (CC = [ št, žd] )




- 1.1.1.3.1.2.1. živo
  - 1.1.1.3.1.2.1.1. mn1 \$164
  - 1.1.1.3.1.2.1.2. mn12 \$165 #dužd
- 1.1.1.3.1.2.2. neživo
  - 1.1.1.3.1.2.2.1. mn1 \$166
  - 1.1.1.3.1.2.2.2. mn12 \$167 #80
- 1.1.1.3.2. jednosložne osnove s dočetakom na -eC
  - 1.1.1.3.2.1. živo
    - 1.1.1.3.2.1.1. mn1 \$168
    - 1.1.1.3.2.1.2. mn12 \$169 #81
  - 1.1.1.3.2.2. neživo
    - 1.1.1.3.2.2.1. mn1 \$170
    - 1.1.1.3.2.2.2. mn12 \$171 #82
- 1.1.1.3.3. dvosložne osnove s dočetakom na -eC
  - 1.1.1.3.3.1. živo
    - 1.1.1.3.3.1.1. mn1 \$172 #83
    - 1.1.1.3.3.1.2. mn12 \$173
  - 1.1.1.3.3.2. neživo
    - 1.1.1.3.3.2.1. mn1 \$174 #84
    - 1.1.1.3.3.2.2. mn12 \$175
- 1.1.1.3.4. dočetak na -aC (a = nepostojano) tj. na -ač, -ađ, -alj, -anj
  - 1.1.1.3.4.1. živo
    - 1.1.1.3.4.1.1. mn1 \$176
    - 1.1.1.3.4.1.2. mn12 \$177 #85
  - 1.1.1.3.4.2. neživo
    - 1.1.1.3.4.2.1. mn1 \$178
    - 1.1.1.3.4.2.2. mn12 \$179 #86
- 1.1.1.4. dočetak na -c
  - 1.1.1.4.1. dočetak na -oc, -uc, -ac (a <> nepostojano a), -rc (r = vokalsko r), -Cc i *stric*
    - 1.1.1.4.1.1. »domaće« s nastavkom u Vjd na -e
      - 1.1.1.4.1.1.1. živo
        - 1.1.1.4.1.1.1.1. mn1 \$180 #87
        - 1.1.1.4.1.1.1.2. mn12 \$181 #mrc
      - 1.1.1.4.1.1.2. neživo
        - 1.1.1.4.1.1.2.1. mn1 \$182 #89
        - 1.1.1.4.1.1.2.2. mn12 \$183
    - 1.1.1.4.1.2. posuđenice s nastavkom u Vjd na -u
      - 1.1.1.4.1.2.1. živo
        - 1.1.1.4.1.2.1.1. mn1 \$184
        - 1.1.1.4.1.2.1.2. mn12 \$185
      - 1.1.1.4.1.2.2. neživo
        - 1.1.1.4.1.2.2.1. mn1 \$186
        - 1.1.1.4.1.2.2.2. mn12 \$187 #90
    - 1.1.1.4.1.3. *stric* \$188 #88
  - 1.1.1.4.2. dočetak na -ec
    - 1.1.1.4.2.1. jednosložne osnove
      - 1.1.1.4.2.1.1. »domaće« s nastavkom u Vjd na -e
        - 1.1.1.4.2.1.1.1. živo
          - 1.1.1.4.2.1.1.1.1. mn1 \$189 #žrec
          - 1.1.1.4.2.1.1.1.2. mn12 \$190 #91
        - 1.1.1.4.2.1.1.2. neživo
          - 1.1.1.4.2.1.1.2.1. mn1 \$191

1.1.1.4.2.1.1.2.2. mn12 \$192 #92  
 1.1.1.4.2.1.2. posuđenice s nastavkom u Vjd na -u  
 1.1.1.4.2.1.2.1. živo  
 1.1.1.4.2.1.2.1.1. mn1 \$193  
 1.1.1.4.2.1.2.1.2. mn12 \$194  
 1.1.1.4.2.1.2.2. neživo  
 1.1.1.4.2.1.2.2.1. mn1 \$195  
 1.1.1.4.2.1.2.2.2. mn12 \$196 #nec  
 1.1.1.4.2.2. višesložne osnove  
 1.1.1.4.2.2.1. »domaće« s nastavkom u Vjd na -e  
 1.1.1.4.2.2.1.1. živo  
 1.1.1.4.2.2.1.1.1. mn1 \$197 #93  
 1.1.1.4.2.2.1.1.2. mn12 \$198 #poluzec  
 1.1.1.4.2.2.1.2. neživo osim mjesec  
 1.1.1.4.2.2.1.2.1. mn1 \$199 #94  
 1.1.1.4.2.2.1.2.2. mn12 \$200  
 1.1.1.4.2.2.1.3. mjesec \$201 #95  
 1.1.1.4.2.2.2. posuđenice s nastavkom u Vjd na -u  
 1.1.1.4.2.2.2.1. živo  
 1.1.1.4.2.2.2.1.1. mn1 \$202  
 1.1.1.4.2.2.2.1.2. mn12 \$203  
 1.1.1.4.2.2.2.2. neživo  
 1.1.1.4.2.2.2.2.1. mn1 \$204 #gablec  
 1.1.1.4.2.2.2.2.2. mn12 \$205 #polunec  
 1.1.1.4.3. dočetak na -ic (osim stric)  
 1.1.1.4.3.1. živo  
 1.1.1.4.3.1.1. mn1 \$206 #96  
 1.1.1.4.3.1.2. mn12 \$207 #97  
 1.1.1.4.3.2. neživo  
 1.1.1.4.3.2.1. mn1 \$208 #98  
 1.1.1.4.3.2.2. mn12 \$209 #99  
 1.1.1.4.4. dočetak na -ac (a = nepostojano)  
 1.1.1.4.4.1. osnova na -Cac (a = nepostojano, C = [f, j, k, l\*, lj, m, n, nj, p, r, š, v] \* (kad je predzadnji slog naglašen dugouzvano)  
 1.1.1.4.4.1.1. živo  
 1.1.1.4.4.1.1.1. mn1 \$210 #100  
 1.1.1.4.4.1.1.2. mn12 \$211 #101  
 1.1.1.4.4.1.2. neživo  
 1.1.1.4.4.1.2.1. mn1 \$212 #102  
 1.1.1.4.4.1.2.2. mn12 \$213 #103  
 1.1.1.4.4.2. (višesložna) osnova na -lac (a = nepostojano, l > o), a znače vršitelja radnje i talac, činilac  
 1.1.1.4.4.2.1. živo \$214 #104  
 1.1.1.4.4.2.2. neživo \$215 #činilac  
 1.1.1.4.4.3. osnova na -sac (a = nepostojano)  
 1.1.1.4.4.3.1. živo  
 1.1.1.4.4.3.1.1. mn1 \$216 #105  
 1.1.1.4.4.3.1.2. mn12 \$217  
 1.1.1.4.4.3.2. neživo  
 1.1.1.4.4.3.2.1. mn1 \$218 #106  
 1.1.1.4.4.3.2.2. mn12 \$219  
 1.1.1.4.4.4. osnova na -tac (a = nepostojano)  
 1.1.1.4.4.4.1. okrnjena osnova u Nmn

- 1.1.1.4.4.4.1.1. živo
  - 1.1.1.4.4.4.1.1.1. mn1 \$220 #107
  - 1.1.1.4.4.4.1.1.2. mn12 \$221 #108
- 1.1.1.4.4.4.1.2. neživo
  - 1.1.1.4.4.4.1.2.1. mn1 \$222 #109
  - 1.1.1.4.4.4.1.2.2. mn12 \$223 #110
- 1.1.1.4.4.4.2. neokrnjena osnova u Nmn
  - 1.1.1.4.4.4.2.1. živo
    - 1.1.1.4.4.4.2.1.1. mn1 \$224
    - 1.1.1.4.4.4.2.1.2. mn12 \$225
  - 1.1.1.4.4.4.2.2. neživo
    - 1.1.1.4.4.4.2.2.1. mn1 \$226
    - 1.1.1.4.4.4.2.2.2. mn12 \$227
- 1.1.1.4.4.5. osnova na -stac i -štac (a = nepostojano)
  - 1.1.1.4.4.5.1. živo
    - 1.1.1.4.4.5.1.1. mn1 \$228 #vještac
    - 1.1.1.4.4.5.1.2. mn12 \$229
  - 1.1.1.4.4.5.2. neživo
    - 1.1.1.4.4.5.2.1. mn1 \$230 #111
    - 1.1.1.4.4.5.2.2. mn12 \$231
- 1.1.1.4.4.6. osnova na -čac i -ćac (a = nepostojano)
  - 1.1.1.4.4.6.1. živo
    - 1.1.1.4.4.6.1.1. mn1 \$232 #112
    - 1.1.1.4.4.6.1.2. mn12 \$233
  - 1.1.1.4.4.6.2. neživo
    - 1.1.1.4.4.6.2.1. mn1 \$234 #113
    - 1.1.1.4.4.6.2.2. mn12 \$235
- 1.1.1.4.4.7. osnova na -Cac (a = nepostojano, C = zvučni tj. [b, d, dž, đ, g, z, ž])
  - 1.1.1.4.4.7.1. osnova na -Cac (C = [b, g, ž])
    - 1.1.1.4.4.7.1.1. živo
      - 1.1.1.4.4.7.1.1.1. mn1 \$236 #114
      - 1.1.1.4.4.7.1.1.2. mn12 \$237
    - 1.1.1.4.4.7.1.2. neživo
      - 1.1.1.4.4.7.1.2.1. mn1 \$238 #115
      - 1.1.1.4.4.7.1.2.2. mn12 \$239
  - 1.1.1.4.4.7.2. osnova na -zac
    - 1.1.1.4.4.7.2.1. živo
      - 1.1.1.4.4.7.2.1.1. mn1 \$240 #116
      - 1.1.1.4.4.7.2.1.2. mn12 \$241
    - 1.1.1.4.4.7.2.2. neživo
      - 1.1.1.4.4.7.2.2.1. mn1 \$242 #117
      - 1.1.1.4.4.7.2.2.2. mn12 \$243
  - 1.1.1.4.4.7.3. osnova na -dac
    - 1.1.1.4.4.7.3.1. okrnjena osnova u Nmn
      - 1.1.1.4.4.7.3.1.1. živo
        - 1.1.1.4.4.7.3.1.1.1. mn1 \$244 #118
        - 1.1.1.4.4.7.3.1.1.2. mn12 \$245
      - 1.1.1.4.4.7.3.1.2. neživo
        - 1.1.1.4.4.7.3.1.2.1. mn1 \$246 #119
        - 1.1.1.4.4.7.3.1.2.2. mn12 \$247
    - 1.1.1.4.4.7.3.2. neokrnjena osnova u Nmn
      - 1.1.1.4.4.7.3.2.1. živo
        - 1.1.1.4.4.7.3.2.1.1. mn1 \$248

- 1.1.1.4.4.7.3.2.1.2. mn12 \$249
- 1.1.1.4.4.7.3.2.2. neživo
  - 1.1.1.4.4.7.3.2.2.1. mn1 \$250
  - 1.1.1.4.4.7.3.2.2.2. mn12 \$251
- 1.1.1.4.4.7.4. osnova na -zdac, -ždac (a = nepostojano)
  - 1.1.1.4.4.7.4.1. živo
    - 1.1.1.4.4.7.4.1.1. mn1 \$252 #duždac
    - 1.1.1.4.4.7.4.1.2. mn12 \$253
  - 1.1.1.4.4.7.4.2. neživo
    - 1.1.1.4.4.7.4.2.1. mn1 \$254 #120
    - 1.1.1.4.4.7.4.2.2. mn12 \$255
- 1.1.1.4.4.7.5. osnova na -đac
  - 1.1.1.4.4.7.5.1. živo
    - 1.1.1.4.4.7.5.1.1. mn1 \$256 #121
    - 1.1.1.4.4.7.5.1.2. mn12 \$257
  - 1.1.1.4.4.7.5.2. neživo
    - 1.1.1.4.4.7.5.2.1. mn1 \$258
    - 1.1.1.4.4.7.5.2.2. mn12 \$259
- 1.1.1.5. dočetak na -o (< l)
  - 1.1.1.5.1. dočetak na -Vo (V <> nepostojano a, o < l) (osim dio i složenica odio/odjel, predio/predjel, podrazdio/podrazdjel, razdio/razdjel, udio/udjel)
    - 1.1.1.5.1.1. živo
      - 1.1.1.5.1.1.1. mn1 \$260 #122
      - 1.1.1.5.1.1.2. mn12 \$261
    - 1.1.1.5.1.2. neživo
      - 1.1.1.5.1.2.1. mn1 \$262 #123
      - 1.1.1.5.1.2.2. mn12 \$263 #124
  - 1.1.1.5.2. dio i složenice odio/odjel, predio/predjel, podrazdio/podrazdjel, razdio/razdjel, udio/udjel
    - 1.1.1.5.2.1. dio \$264 #125
    - 1.1.1.5.2.2. odio/odjel, predio/predjel, podrazdio/podrazdjel, razdio/razdjel, udio/udjel \$265 #126
  - 1.1.1.5.3. alternativni dočetak na -ô/-ol
    - 1.1.1.5.3.1. živo
      - 1.1.1.5.3.1.1. mn1 \$266 #127
      - 1.1.1.5.3.1.2. mn12 \$267
    - 1.1.1.5.3.2. neživo
      - 1.1.1.5.3.2.1. mn1 \$268 #128
      - 1.1.1.5.3.2.2. mn12 \$269
  - 1.1.1.5.4. dočetak na -ao (a = nepostojano)
    - 1.1.1.5.4.1. živo
      - 1.1.1.5.4.1.1. mn1 \$270 #đavao
      - 1.1.1.5.4.1.2. mn12 \$271 #129
    - 1.1.1.5.4.2. neživo
      - 1.1.1.5.4.2.1. mn1 \$272 #131
      - 1.1.1.5.4.2.2. mn12 \$273 #132
- 1.1.1.6. dočetak na -V
  - 1.1.1.6.1. dočetak na -ô, -e, -u
    - 1.1.1.6.1.1. živo mn1 \$274 #133
    - 1.1.1.6.1.2. neživo mn1 \$275 #134
  - 1.1.1.6.2. dočetak na -i, -y
    - 1.1.1.6.2.1. živo mn1 \$276 #135
    - 1.1.1.6.2.2. neživo mn1 \$277 #136

- 1.1.1.7. posebni i pojedinačni uzorci imenica koje se svršavaju na C(prednepčani-c) (*car*; *gospodar*; na -âr a znače zanimanje (?); na -îr, -er a znače živo; -êr, etnici i tuđice na -êz, -îz, -ûz; *knez*; *vitez*; *markiz*; *put*)
  - 1.1.1.7.1. *car* \$278 #137
  - 1.1.1.7.2. *gospodar* \$279 #138
  - 1.1.1.7.3. na -âr
    - 1.1.1.7.3.1. živo
      - 1.1.1.7.3.1.1. mn1 \$280 #139
      - 1.1.1.7.3.1.2. mn12 \$281
    - 1.1.1.7.3.2. neživo
      - 1.1.1.7.3.2.1. mn1 \$282
      - 1.1.1.7.3.2.2. mn12 \$283 #kotar
  - 1.1.1.7.4. imenice na -îr, -er
    - 1.1.1.7.4.1. živo
      - 1.1.1.7.4.1.1. mn1 \$284 #140
      - 1.1.1.7.4.1.2. mn12 \$285 #141
    - 1.1.1.7.4.2. neživo
      - 1.1.1.7.4.2.1. mn1 \$286
      - 1.1.1.7.4.2.2. mn12 \$287 #cer (crni hrast)
  - 1.1.1.7.5. imenice na -êr, etnici i tuđice na -êz, -îz, -ûz 
    - 1.1.1.7.5.1. živo
      - 1.1.1.7.5.1.1. mn1 \$288 #142
      - 1.1.1.7.5.1.2. mn12 \$289 #pêr
    - 1.1.1.7.5.2. neživo
      - 1.1.1.7.5.2.1. mn1 \$290 #francuz (ključ)
      - 1.1.1.7.5.2.2. mn12 \$291 #bluz
  - 1.1.1.7.6. *knez* \$292 #143
  - 1.1.1.7.7. *vitez* \$293 #144
  - 1.1.1.7.8. *markiz* \$294
  - 1.1.1.7.9. *put* \$295 #145
- 1.1.2. imenice m s -o/-e
  - 1.1.2.1. s nastavkom -o
    - 1.1.2.1.1. osnova na -k (osim *dečko*)
      - 1.1.2.1.1.1. živo \$296 #146
      - 1.1.2.1.1.2. neživo \$297 #široko (vjetar)
    - 1.1.2.1.2. *dečko* \$298 #147
    - 1.1.2.1.3. osnova na -C(prednepčani) tj. [p, b, v, f, m, n, t, d, s, z, l, r]
      - 1.1.2.1.3.1. živo \$299 #148
      - 1.1.2.1.3.2. neživo \$300
  - 1.1.2.2. s nastavkom -e i dočetakom -C(nepčani)
    - 1.1.2.2.1. živo \$301 #149
    - 1.1.2.2.2. neživo \$302
- 1.1.3. tuđice m s -V koji se vlada kao nastavak
  - 1.1.3.1. s nastavkom -o
    - 1.1.3.1.1. osnova na -C osim [k, g] ili samoglasnik osim [i]
      - 1.1.3.1.1.1. živo \$303 #150
      - 1.1.3.1.1.2. neživo \$304 #151
    - 1.1.3.1.2. osnova na -k
      - 1.1.3.1.2.1. dvosložne imenice
        - 1.1.3.1.2.1.1. živo \$305
        - 1.1.3.1.2.1.2. neživo \$306 #152
      - 1.1.3.1.2.2. višesložne imenice

- 1.1.3.1.2.2.1. živo \$307 #tehniko
- 1.1.3.1.2.2.2. neživo \$308 #153
- 1.1.3.1.3. osnova na -g
  - 1.1.3.1.3.1. dvosložne imenice
    - 1.1.3.1.3.1.1. živo \$309 #154
    - 1.1.3.1.3.1.2. neživo \$310
  - 1.1.3.1.3.2. višesložne imenice
    - 1.1.3.1.3.2.1. živo \$311 #155
    - 1.1.3.1.3.2.2. neživo \$312
- 1.1.3.1.4. osnova na -i
  - 1.1.3.1.4.1. živo \$313 #156
  - 1.1.3.1.4.2. neživo \$314 #157
- 1.1.3.2. s nastavkom -e
  - 1.1.3.2.1. živo \$315 #158
  - 1.1.3.2.2. neživo \$316 #159
- 1.1.3.3. s nastavkom -i
  - 1.1.3.3.1. živo \$317
  - 1.1.3.3.2. neživo \$318 #160
- 1.2. imenice n
  - 1.2.1. imenice n sa -o (osim *oko* i *uho*)
    - 1.2.1.1. jednakosložne
      - 1.2.1.1.1. osnova na -VC ili -VCC (CC = [ st, zd] ) i *kopno*
        - 1.2.1.1.1.1. osnova na -C (C = prednepčani glas tj. [ p, b, v, f, m, n, t, d, s, z, l, r] ) ili -CC (CC = [ st, zd] ) i *kopno* \$319 #161
        - 1.2.1.1.1.2. osnova na -k, -g, -h \$320 #162
      - 1.2.1.1.2. osnova na -CC (CC <> [ st, zd] i C1 se ne obezvučuje) osim *kopno* \$321 #164
      - 1.2.1.1.3. osnova na -CC (CC <> [ st, zd] i C1 se obezvučuje) \$322 #165
    - 1.2.1.2. nejednakosložne (samo u množini)
      - 1.2.1.2.1. osnova na -s (*čudo, kolo, nebo, tijelo*) \$323 #166
      - 1.2.1.2.2. osnova na -t (*drvo*) \$324 #167
  - 1.2.2. imenice n sa -e
    - 1.2.2.1. jednakosložne sa dočetakom na -Ce (C = prednjonepčani\* ili -c-) ili -CCe (CC = [ št, šć, žđ] ili C2 = [ j, lj] ) i *more, tle* (\* osim *janje, teneće, sirće*)
      - 1.2.2.1.1. dočetak na -VCE ili -CCe (CC = [ št, šć, žđ] ili C2 = [ j, lj] osim *koplje* \$325 #168
      - 1.2.2.1.2. dočetak na -CCe (CC <> [ št, šć, žđ] i C2 <> [ j, lj] ) i *koplje* (Gmn proširena osnova s nepostojanim a)
        - 1.2.2.1.2.1. dočetak na -CCe (C1 = sonant tj. [ f, v, l, lj, r, m, n, nj, j] i C2 <> [ c] ) i *koplje, vrhnje, tle* \$326 #koplje #vrhnje #tle
    - 1.2.2.1.2.2. dočetak na -Cce
      - 1.2.2.1.2.2.1. dočetak na -Cce \$327 #169
      - 1.2.2.1.2.2.2. osnova na -Cce (C1 se u Gmn vraća u prvotno zvučno stanje) \$328 #170
      - 1.2.2.1.2.2.3. osnova na -Cce (C1 se u Gmn vraća, u ostalim padežima ispao)
        - 1.2.2.1.2.2.3.1. vraća se -d- \$329 #171
        - 1.2.2.1.2.2.3.2. vraća se -t- \$330 #172
      - 1.2.2.1.2.2.4. osnova na -lce (l > o) \$331 #173
  - 1.2.2.2. nejednakosložne sa dočetakom na -C(prednepčani) (osim *more, tle, finale?*) ili -C(srednjonepčani) i *jaje, janje, teneće, sirće*)
    - 1.2.2.2.1. osnova na -t

- 1.2.2.2.1.1. živo osim *dijete* \$332 #174
- 1.2.2.2.1.2. *dijete* \$333 #175
- 1.2.2.2.1.3. neživo osim *jaje* \$334 #176
- 1.2.2.2.1.4. *jaje* \$335
- 1.2.2.2.2. osnova na -n \$336 #177
- 1.2.2.2.3. podne, dopodne, popodne, prijepodne, poslijepodne \$337 #178
- 1.2.2.2.4. *veče* \$338 #179
- 1.2.3. *oko* \$339 #180
- 1.2.4. *uho* \$340 #181
- 1.2.5. *doba* \$341 #182
- 1.2.6. *vrata, usta, pluća* \$342

## 4.2.2. e-vrsta

### 2. e-vrsta

#### 2.1. imenice s -a

- 2.1.1. imenice sa dočetakom -VCa\* ili -VCCa (CC = [ st, št, zd, žd, šč, šćj) (\* osim tro- ili višesložnih na -ica)
  - 2.1.1.1. osnovni se C u DLjd ne mijenja
    - 2.1.1.1.1. s jednim nastavkom u Vjd -o \$343 #183
    - 2.1.1.1.2. s jednim nastavkom u Vjd -a \$344 #mama #tata
    - 2.1.1.1.3. s dvojnim nastavcima u Vjd -o/-a \$345 #184
  - 2.1.1.2. osnovni se C u DLjd mijenja
    - 2.1.1.2.1. osnova na -k \$346 #185
    - 2.1.1.2.2. osnova na -g \$347 #186
    - 2.1.1.2.3. osnova na -h \$348 #187
    - 2.1.1.2.4. dvojni likovi u Gmn (*ruka, noga, sluga*)
      - 2.1.1.2.4.1. *ruka* \$349 #188
      - 2.1.1.2.4.2. *noga* \$350 #189
      - 2.1.1.2.4.3. *sluga* \$351 #190
    - 2.1.1.2.5. osnova na -k s dvojnim likovima u DLjd \$352 #191
    - 2.1.1.2.6. osnova na -g s dvojnim likovima u DLjd \$353 #192
    - 2.1.1.2.7. osnova na -h s dvojnim likovima u DLjd \$354 #193
  - 2.1.2. tro- ili višesložne imenice na -ica
    - 2.1.2.1. s jednim nastavkom u Vjd -e \$355 #194
    - 2.1.2.2. s dvojnim nastavcima u Vjd -o/-e \$356 #195
- 2.1.3. imenice sa dočetakom -CCa, -CCCa
  - 2.1.3.1. C se u DLjd ne mijenja
    - 2.1.3.1.1. osnova na -CC (C2 = [ l, lj, m, r, v]) s nepostojanim a u Gmn \$357 #196
    - 2.1.3.1.2. osnova na -Ck s dvojnim likovima u Vjd i nepostojanim a u Gmn \$358 #197
    - 2.1.3.1.3. osnova na -Cnj ili -CC (CC = [ kt, nc, nč, nt, rt, sn, db, džb, žb] s trojnim likovima u Gmn \$359 #198
    - 2.1.3.1.4. osnova na -CC (C1 se u Gmn vraća u prvotno zvučno stanje) s trojnim likovima u Gmn \$360 #199
    - 2.1.3.1.5. osnova na -Cnj ili -CC s dvojnim likovima u Gmn \$361 #200
    - 2.1.3.1.6. *kćerca* \$362 #201
    - 2.1.3.1.7. osnova na -CC s trojnim likovima u Gmn i dvojnim u Vjd \$363 #202
    - 2.1.3.1.8. *breskva* \$364 #238

- 2.1.3.2.C se u DLjd mijenja (jedan lik)
  - 2.1.3.2.1. jedan lik u Gmn (nepostojano a) \$365 #203
  - 2.1.3.2.2. dva lika u Gmn (bez nepostojanoga a) \$366 #204
  - 2.1.3.2.3. tri lika u Gmn \$367 #205
  - 2.1.3.2.4. osnova na -CC (C1 se u Gmn vraća u prvotno zvučno stanje) s trojnim likovima u Gmn \$368 #206
- 2.1.3.3.C se u DLjd mijenja i ne mijenja (dvojni likovi)
  - 2.1.3.3.1. tri lika u Gmn \$369 #209
  - 2.1.3.3.2. dva lika u Gmn (bez nepostojanoga a) \$370 #210
  - 2.1.3.3.3. jedan lik u Gmn (nepostojano a) \$371 #211
  - 2.1.3.3.4. dočetak na -CCa (CC = [tk], C1 se u Gmn ne vraća u prvotno zvučno stanje) i trojni likovi u Gmn (*krletka, odgonetka, zagonetka*) \$372 #212
  - 2.1.3.3.5. dočetak na -CCa (CC = [tk], C1 se u Gmn vraća u prvotno zvučno stanje) i dvojni likovi u Gmn \$373 #213
- 2.2.imenice s -e \$374 #214
- 2.3.imenice s -o \$375 #215
- 2.4.imenice s -0 \$376 #216
- 2.5.imenica *mati* \$377 #237

#### 4.2.3. i-vrsta

##### 3. i-vrsta

- 3.1.imenice s dočekom na -C
  - 3.1.1.imenice s nepromjenjivom osnovom (C = [b, č, ć, đ, lj, m, p, r, s, š, v, z, ž])
    - 3.1.1.1.u Ijd nastavci -i/-ju (C = [č, r, s, š, z, ž]) \$378 #217
    - 3.1.1.2.u Ijd nastavci -i/-u (C = [ć, đ, lj]) \$379 #218
    - 3.1.1.3.u Ijd nastavak -i \$380 #219
    - 3.1.1.4.u Ijd nastavci -i/-ju i Gmn -i/-iju (*kokoš, grudi, pleći, prsi*) \$381 #220
    - 3.1.1.5.u Ijd nastavci -i/-lju (C = [b, m, p, v]) \$382 #221
  - 3.1.2.imenice s promjenjivom osnovom (C = [d, l, n, t] ili CC = [st])
    - 3.1.2.1.bez nepostojanoga a
      - 3.1.2.1.1. osnova na -d (2Ijd -đu) \$383 #222
      - 3.1.2.1.2. osnova na -d (2Ijd -đu), dvojni oblici u DLI mn (zbirna?) \$384 #223
      - 3.1.2.1.3. osnova na -l (2Ijd -lju) \$385 #224
      - 3.1.2.1.4. osnova na -n (2Ijd -nj) \$386 #225
      - 3.1.2.1.5. osnova na -Vt (2Ijd -ću) \$387 #226
      - 3.1.2.1.6. osnova na -st(-zd) (2Ijd -šću, -žđu) \$388 #227
      - 3.1.2.1.7. *kost* \$389 #228
    - 3.1.2.2.s nepostojanim a
      - 3.1.2.2.1. osnova na -VCC (C1 <> [s, z]) \$390 #229
      - 3.1.2.2.2. osnova na -VsC \$391 #230
      - 3.1.2.2.3. osnova na -VzC \$392 #231
- 3.2.imenice s dočekom na -V
  - 3.2.1.imenice na -ao (a = nepostojano, o < l)
    - 3.2.1.1. osnova na -Cl (C se ne mijenja) \$393 #232
    - 3.2.1.2. osnova na -sl \$394 #233
    - 3.2.1.3. osnova na -zl \$395 #234
    - 3.2.1.4. osnova na -stl (+ -št, -zd, -žd) \$396 #235



3.2.2. imenica na -i (kći) \$397 #236

#### 4.2.4. Poimeničeni pridjevi

##### 4. pridjevska deklinacija imenica<sup>116</sup>

###### 4.1. imenice m

4.1.1. osnova -C (C <> nepčani)

4.1.1.1. živo \$398 #stari

4.1.1.2. neživo \$399

4.1.2. osnova -C (C = nepčani tj. [ć, đ, š, ž, č, dž, j, lj, nj]) ili  
-CC (CC = [št, žd])

4.1.2.1. živo \$400 #Smeđi

4.1.2.2. neživo \$401

###### 4.2. imenice f \$402 #Hrvatska

###### 4.3. imenice n

4.3.1. osnova -C (C <> nepčani) \$403 #slatko

4.3.2. osnova -C (C = nepčani tj. [ć, đ, š, ž, č, dž, j, lj, nj]) ili  
-CC (CC = [št, žd]) \$404

#### 4.2.5. Pridjevska deklinacija

##### 4.2.5.1. Neodređeni i određeni oblik

##### 5. pridjevska deklinacija

###### 5.1. neodređeni

5.1.1. bez glasovnih promjena \$451



5.1.2. s glasovnim promjenama

5.1.2.1. samo s nepostojanim a \$453

5.1.2.2. jednačenje po zvučnosti \$454

5.1.2.3. gubljenje suglasnika \$456

5.1.2.4. alternacija l/o

5.1.2.4.1. bez drugih promjena \$458

5.1.2.4.2. s nepostojanim a \$459

5.1.2.4.3. s alternacijom ije/i \$460

###### 5.2. određeni

5.2.1. Gjd mr. -og(a)

5.2.1.1. bez glasovnih promjena \$452

5.2.1.2. s glasovnim promjenama

5.2.1.2.1. jednačenje po zvučnost \$455

5.2.1.2.2. gubljenje suglasnika \$457

5.2.1.3. nepotpuna paradigma \$461

5.2.2. Gjd mr. -eg(a) \$462

###### 5.3. brojevi

5.3.1. dva, oba, obadva

---

<sup>116</sup> O uključivanju poimeničenih pridjeva u imeničnu deklinaciju v. Tadić (1992:12).

- 5.3.1.1. dva, obadva \$486
- 5.3.1.2. oba \$487
- 5.3.2. tri, četiri
  - 5.3.2.1. tri \$488
  - 5.3.2.2. četiri \$489
- 5.3.3. zbirne imenice
  - 5.3.3.1. dvoje \$490
  - 5.3.3.2. troje \$491
  - 5.3.3.3. četvoro/četvero \$492

#### **4.2.5.2. Zamjenička deklinacija**

- 6. zamjenička deklinacija
  - 6.1. imeničke zamjenice
    - 6.1.1. lične zamjenice
      - 6.1.1.1. ja, ti, sebe
        - 6.1.1.1.1. ja \$463
        - 6.1.1.1.2. ti \$464
        - 6.1.1.1.3. sebe \$468
      - 6.1.1.2. mi, vi
        - 6.1.1.2.1. mi \$465
        - 6.1.1.2.2. vi \$466
      - 6.1.1.3. on, ona, ono
    - 6.1.2. neodređene imeničke zamjenice
      - 6.1.2.1. tko, što, ništa
        - 6.1.2.1.1. tko
        - 6.1.2.1.2. što
        - 6.1.2.1.3. ništa
      - 6.1.2.2. tkogod, štogod
        - 6.1.2.2.1. tkogod
        - 6.1.2.2.2. štogod
  - 6.2. pridjevine zamjenice
    - 6.2.1. posvojne zamjenice
      - 6.2.1.2. moj, tvoj, svoj \$476
      - 6.2.1.2. naš, vaš \$477
    - 6.2.2. pokazne zamjenice
      - 6.2.2.1. taj \$478
      - 6.2.2.2. ovaj \$479
      - 6.2.2.3. sam \$480
    - 6.2.3. neodređene pridjevine zamjenice
      - 6.2.3.1. koji, čiji i složenice
        - 6.2.3.1.1. bez -god
          - 6.2.3.1.1.1. koji \$481
          - 6.2.3.1.1.2. čiji i složenice \$482
        - 6.2.3.1.2. s -god
          - 6.2.3.1.2.1. koji \$483
          - 6.2.3.1.2.2. čiji i složenice \$484
      - 6.2.3.2. sav \$485

## 4.3. Konjugacija

### 4.3.1. Neplodni glagoli

#### 1. neplodni glagoli

##### 1.1. Prvi razred (-Vsti, -nijeti)

###### 1.1.1. infinitiv na -Vsti

###### 1.1.1.1. osnova sa t, d (t, d ispadaju u prad)

###### 1.1.1.1.1. prezentska osnova sa -n-

###### 1.1.1.1.1.1. nesvršeni

1.1.1.1.1.1.1. impf. na -ijah \$551

1.1.1.1.1.1.1. osnova na -t (impf. -tah) \$552???

1.1.1.1.1.1.2. osnova na -d (impf. -đah) \$553

1.1.1.1.1.2. svršeni \$554 #pasti/padnem, sjesti/sjednem, sresti/sretnem

###### 1.1.1.1.2. prezentska osnova bez -n-

###### 1.1.1.1.2.1. o u osnovi (prad sa stegnutim o)

###### 1.1.1.1.2.1.1. nesvršeni

1.1.1.1.2.1.1.1. impf. na -ijah \$555 #bosti/bo/bodijah

1.1.1.1.2.1.1.2. osnova na -t (impf. na -tah) \$556?

1.1.1.1.2.1.1.3. osnova na -d (impf. na -đah) \$557  
#bosti/bođah

1.1.1.1.2.1.2. svršeni \$558 #zabosti/zabo/zabodavši, zabodav, zabovši

###### 1.1.1.1.2.2. u osnovi ostali samoglasni (prad bez stegnutoga o)

###### 1.1.1.1.2.2.1. nesvršeni

1.1.1.1.2.2.1.1. impf. na -ijah \$559  
#plesti/pleo/pletijah

1.1.1.1.2.2.1.2. osnova na -t (impf. -tah) \$560  
#plesti/pletah

1.1.1.1.2.2.1.3. osnova na -d (impf. -đah) \$561  
#jesti/jedah

1.1.1.1.2.2.2. svršeni \$562 #splesti/spleo/spletavši, spletav, splevši

###### 1.1.1.2. osnova sa s, z (s, z ne ispadaju u prad)

###### 1.1.1.2.1. nesvršeni

1.1.1.2.1.1. impf. na -ijah \$563 #gristi/grizao/grizijah

1.1.1.2.1.2. impf. na -ah \$564 #gristi/grizah

1.1.1.2.2. svršeni \$565 #zagristi/zagrizao/zagrizavši, zagrizav

###### 1.1.1.3. osnova sa st (zd?)

###### 1.1.1.3.1. nesvršeni

1.1.1.3.1.1. impf. na -ijah \$566 #rasti/rastao/rastijah

1.1.1.3.1.2. impf. na -ah \$567 #rasti/rastao/rastah

1.1.1.3.2. svršeni \$568 #narasti/narastao/narastavši, narastav

###### 1.1.2. infinitiv na -nijeti/-nesem \$569 #donijeti/donesem

##### 1.2. Drugi razred (-Csti)

###### 1.2.1. nesvršeni


1.2.1.1. impf. na -ijah \$570 #grepsti/grebem/grebijah

1.2.1.2. impf. na -ah \$571 #grepsti/grebem/grebah

1.2.2. svršeni \$572 #izgrepsti/izgrebem

##### 1.3. Treći razred (-ći)

###### 1.3.1. dočetak na k, g, (h?)

- 1.3.1.1.1. prezentska osnova sa -n-
  - 1.3.1.1.1.1. aor. s -n- \$573 #pući/puknem/puknuh
  - 1.3.1.1.1.2. aor. bez -n-
    - 1.3.1.1.1.2.1. impr. sa -n- \$574 #pući/puknem/pukoh
    - 1.3.1.1.1.2.2. impr. bez -n- \$575 # (po)moći/ (po)mognem/ (po)mogoh/pomozi, leći/lezi, vrći/vrzi, pregni/prezi
  - 1.3.1.1.1.3. alternativni prezent \$576 #reći/rečem/rekoh
- 1.3.1.2. prezentska osnova bez -n-
  - 1.3.1.2.1. prvo lice prez. na -u #moći/mogu \$577
  - 1.3.1.2.2. prvo lice prez. na -m
    - 1.3.1.2.2.1. nesvršeni
      - 1.3.1.2.2.1.1. impf. na -ijah #peći/pečem/pecijah \$578
      - 1.3.1.2.2.1.2. impf. na -ah #peći/pečem/pečah \$579
    - 1.3.1.2.2.2. svršeni #upeći/upečem/upekoh \$580
- 1.3.2. dočetak na d, (t)
  - 1.3.2.1. ići \$581
  - 1.3.2.2. otići \$582 #otići/odem/odoh/otišao
  - 1.3.2.3. ostale složenice od ići \$583 #ući/uđem/uđoh/ušao 
- 1.4. Četvrti razred (-Crěti/-Cr̥ti)
  - 1.4.1. inf. dočetak -r̥ti
    - 1.4.1.1. nesvršeni
      - 1.4.1.1.1. dočetak -Carem \$584 #tr̥ti/tarem/trven
      - 1.4.1.1.2. dočetak -Crem \$585 #tr̥ti/trem/trt
    - 1.4.1.2. svršeni
      - 1.4.1.2.1. dočetak -Carem \$586 #str̥ti/starem/strven
      - 1.4.1.2.2. dočetak -Crem \$587 #str̥ti/strem/str̥t
  - 1.4.2. inf. dočetak -rijeti
    - 1.4.2.1. prez. -rijem \$588 #mrijeti/mrijem/mrah
    - 1.4.2.2. prez. -rem
      - 1.4.2.2.1. nesvršeni \$589 #mrijeti/mrem, ždrijeti/ždrem
      - 1.4.2.2.2. svršeni \$590 #umrijeti/umrem, prostrijeti/prostrem
  - 1.4.3. inf. dočetak -reti
    - 1.4.3.1. prez. -rim
      - 1.4.3.1.1. nesvršeni \$591 #vreti/vrim, zreti/zrim
      - 1.4.3.1.2. svršeni \$592 #provreti/provr̥m
    - 1.4.3.2. prez. -rijem
      - 1.4.3.2.1. nesvršeni \$593 #vreti/vrijem, zreti/zrijem
      - 1.4.3.2.2. svršeni \$594 #provreti/provrijem
    - 1.4.3.3. prez. -rem
      - 1.4.3.3.1. nesvršeni \$595 #zreti/zrem
      - 1.4.3.3.2. svršeni \$596 #prozreti/prozrem
- 1.5. Peti razred
  - 1.5.1. prez. osn. -C<sub>1</sub>C<sub>2</sub>-
    - 1.5.1.1. inf. -eti
      - 1.5.1.1.1. nesvršeni \$597 #žeti/žmem
      - 1.5.1.1.2. svršeni \$598 #uzeti/uzmem, oteti, početi, zapeti
    - 1.5.1.2. inf. -jeti (-dj̥eti < -d̥ēdn-)
      - 1.5.1.2.1. aor. -eh, ptrp. -ut \$599 #dj̥eti/dj̥eh/dj̥enuo/dj̥enut
      - 1.5.1.2.2. aor. -doh, ptrp. -ven \$600 #dj̥eti/djedoh/djeo/djeven
    - 1.5.1.3. inf. -uti
      - 1.5.1.3.1. nesvršeni
        - 1.5.1.3.1.1. ptrp. -ut \$654 #duti/dmem/dut
        - 1.5.1.3.1.2. ptrp. -uven \$655 #duti/dujem/duven


- 1.5.1.3.2. svršeni
  - 1.5.1.3.2.1. ptrp. -ut \$601 #naduti/nadmem/nadut, usuti/uspem/usut
  - 1.5.1.3.2.2. ptrp. -uven \$602 #naduti/nadmem/naduven
- 1.5.2. prez. osn. -C<sub>1</sub>VC<sub>2</sub>-
  - 1.5.2.1. prezent -im
    - 1.5.2.1.1. nesvršeni \$603 #bdjeti/bdim
    - 1.5.2.1.2. svršeni \$604 #probdjeti/probdim
  - 1.5.2.2. prez. -em
    - 1.5.2.2.1. ptrp. na -ven
      - 1.5.2.2.1.1. nesvršeni \$605 #bdjeti/bdijem, čuti/čujem, smjeti/smijem
      - 1.5.2.2.1.2. svršeni \$606 #probdjeti/probdijem, čuti/čujem, obuti/obuven, izuti/izuven
    - 1.5.2.2.2. ptrp. na -t
      - 1.5.2.2.2.1. nesvršeni \$607
      - 1.5.2.2.2.2. svršeni \$608 #obuti/obut, izuti/izut
    - 1.5.2.2.3. ptrp. na -ljen
      - 1.5.2.2.3.1. nesvršeni \$609 #umjeti/umijem
      - 1.5.2.2.3.2. svršeni \$610 #razumjeti/razumijem
- 1.5.3. ostali glagoli
  - 1.5.3.1. kleti/kunem i složenice
    - 1.5.3.1.1. nesvršeni \$611 #kleti/kunem
    - 1.5.3.1.2. svršeni \$612 #prokleti/prokunem
  - 1.5.3.2. mljeti/meljem i složenice
    - 1.5.3.2.1. nesvršeni \$613 #mljeti/meljem
    - 1.5.3.2.2. svršeni \$614 #samljeti/sameljem
  - 1.5.3.3. žeti/žanjem i složenice
    - 1.5.3.3.1. žeti/žanjem
      - 1.5.3.3.1.1. nesvršeni \$615 #žeti/žanjem
      - 1.5.3.3.1.2. svršeni \$616 #požeti/požanjem
    - 1.5.3.3.2. žnjeti/žnjem i složenice
      - 1.5.3.3.2.1. nesvršeni \$617 #žnjeti/žnjem
      - 1.5.3.3.2.2. svršeni \$618 #požnjeti/požnjem
  - 1.5.3.4. htjeti
    - 1.5.3.4.1. htjeti/ću \$619
    - 1.5.3.4.2. hotjeti/hoću \$620
    - 1.5.3.4.3. -htjeti/-htijem \$621
    - 1.5.3.4.4. htjeti/htjednem \$622
    - 1.5.3.4.5. negirani oblici \$623
- 1.6. Šesti razred
  - 1.6.1. prez. -jem
    - 1.6.1.1. ptrp. -ven
      - 1.6.1.1.1. nesvršeni \$624 #liti/lijem/liven
      - 1.6.1.1.2. svršeni \$625 #proliti/prolijem/proliven
    - 1.6.1.2. ptrp. -jen
      - 1.6.1.2.1. nesvršeni \$626 #piti/pijem/pijen
      - 1.6.1.2.2. svršeni \$627 #popiti/popijem/popijen
    - 1.6.1.3. ptrp. -it
      - 1.6.1.3.1. nesvršeni \$628 #liti/lijem/lit
      - 1.6.1.3.2. svršeni \$629 #proliti/prolijem/prolit
  - 1.6.2. prez. -im
    - 1.6.2.1. nesvršeni \$630 #sniti/snim
    - 1.6.2.2. svršeni \$631 #usniti/usnim



- 1.6.3.bititi
  - 1.6.3.1.bititi/jesam \$632
  - 1.6.3.2.bititi/sam \$633
  - 1.6.3.3.bititi/budem \$634
  - 1.6.3.4.bititi/budnem \$635
- 1.7.Sedmi razred
  - 1.7.1.prez. na -am
    - 1.7.1.1.bez promjena na dočetu osnovne
      - 1.7.1.1.1.nesvršeni \$636 #sjati
      - 1.7.1.1.2.svršeni \$637 #zasjati
    - 1.7.1.2.skraćena prez. osnovne
      - 1.7.1.2.1.nesvršeni \$638 #znati/znam
      - 1.7.1.2.2.svršeni
        - 1.7.1.2.2.1. skraćena osn. u aor. \$639 #dati/dam/dah
        - 1.7.1.2.2.2. neskrraćena osn. u aor. \$640 #-zdati/-zdam/-zdaoh
  - 1.7.2.prez. na -em
    - 1.7.2.1.e kao dočetni samoglas u osnovi
      - 1.7.2.1.1.proširena prez. osn.
        - 1.7.2.1.1.1.dočetak prez. osn. -ade
          - 1.7.2.1.1.1.1. nesvršeni \$641 #znati/znadem
          - 1.7.2.1.1.1.2. svršeni \$642 #dati/dadem
        - 1.7.2.1.1.2.dočetak prez. osn. -adne \$643 #dati/dadnem
        - 1.7.2.1.1.3.dočetak prez. osn. -ne
          - 1.7.2.1.1.3.1. skraćena osn. u aor. \$644 #stati/stanem/stah
          - 1.7.2.1.1.3.2. neskrraćena osn. u aor. \$645 #stati/stanem/stadoh
      - 1.7.2.1.2.neproširena prez. osn.
        - 1.7.2.1.2.1. nesvršeni \$646 #brati
        - 1.7.2.1.2.1. svršeni \$647 #ubрати
    - 1.7.2.2.ě kao dočetni samoglas u osnovi
      - 1.7.2.2.1.bez jednačenja u osn.
        - 1.7.2.2.1.1. nesvršeni \$648 #klati
        - 1.7.2.2.1.2. svršeni \$649 #poklati
      - 1.7.2.2.2.sa jednačenjem u osn.
        - 1.7.2.2.2.1. nesvršeni \$650 #slati/šaljem
        - 1.7.2.2.2.2. svršeni \$651 #poslati/pošaljem
  - 1.7.3.prez. na -im
    - 1.7.3.1.nesvršeni \$652 #spati/spim
    - 1.7.3.2.svršeni \$653 #zaspati/zaspim


## 4.3.2. Plodni glagoli

- 2. plodni glagoli
  - 2.1.a dočetni samoglasnik u osnovi
    - 2.1.1.nesvršeni \$501 #pričati
    - 2.1.2.svršeni \$502 #ispričati
  - 2.2.i dočetni samoglasnik u osnovi
    - 2.2.1.pred i st > št u ptrp
      - 2.2.1.1. nesvršeni
        - 2.2.1.1.1. impf. bez -ijah \$503 #krstiti
        - 2.2.1.1.2. impf s -ijah \$504 #krstiti

- 2.2.1.2. svršeni \$505 #pokrstiti
- 2.2.2. pred i st, zd > šć, žđ u ptrp
  - 2.2.2.1. nesvršeni
    - 2.2.2.1.1. impf. bez -ijah \$506 #gostiti
    - 2.2.2.1.2. impf s -ijah \$507 #gostiti
  - 2.2.2.2. svršeni \$508 #ugostiti
- 2.2.3. pred i bilo što drugo
  - 2.2.3.1. nesvršeni
    - 2.2.3.1.1. impf. bez -ijah \$509 #piliti 
    - 2.2.3.1.2. impf s -ijah \$510 #piliti
  - 2.2.3.2. svršeni \$511 #ispiliti
- 2.3. ě dočetni samoglasnik u osnovi
  - 2.3.1. nesvršeni
    - 2.3.1.1. impf. bez -ijah \$512 #vidjeti
    - 2.3.1.2. impf. s -ijah \$513 #vidjeti
  - 2.3.2. svršeni \$514 #previdjeti
- 2.4. e dočetni samoglasnik u osnovi
  - 2.4.1. ispred e samoglasnik (oe-, ie-, ee-)
    - 2.4.1.1. nesvršeni \$515 #darovati, kazivati, bojovati
    - 2.4.1.2. svršeni \$516
  - 2.4.2. ispred e suglasnik
    - 2.4.2.1. dočetak CpalC-e, prez. -im
      - 2.4.2.1.1. nesvršeni \$517 #vrištati
      - 2.4.2.1.2. svršeni \$518 #zavrištati
    - 2.4.2.2. dočetak C-e, prez. -im
      - 2.4.2.2.1. impr. na -i
        - 2.4.2.2.1.1. nesvršeni \$519 #kričati
        - 2.4.2.2.1.2. svršeni \$520 #zakričati
      - 2.4.2.2.2. impr. na -o
        - 2.4.2.2.2.1. bez promjene korijenskoga samoglasa #bojati se, pojati
          - 2.4.2.2.2.1.1. nesvršeni \$521 #pojati
          - 2.4.2.2.2.1.2. svršeni \$522 #zapojati
        - 2.4.2.2.2.2. promjena korijenskoga samoglasa \$523 #stajati/stojim
    - 2.4.2.3. dočetak sk-e > št-e #iskati, biskati
      - 2.4.2.3.1. sk > št
        - 2.4.2.3.1.1. nesvršeni \$524 #iskati, ištem
        - 2.4.2.3.1.2. svršeni \$525 #zaiskati, zaištem
      - 2.4.2.3.2. sk > šć
        - 2.4.2.3.2.1. nesvršeni \$526 #iskati, išćem
        - 2.4.2.3.2.2. svršeni \$527 #zaiskati, zaišćem
    - 2.4.2.4. dočetak n-e, prez. -nem
      - 2.4.2.4.1. nesvršeni
        - 2.4.2.4.1.1. prad. na -nuo \$528 #sahnuti/sahnuo
        - 2.4.2.4.1.2. prad. i na -(a)o \$529 #sahnuti/sahao
      - 2.4.2.4.2. svršeni
        - 2.4.2.4.2.1. aorist na -nuh
          - 2.4.2.4.2.1.1. dočetak osnove na -dne-, -tne-, -bne-, -pne- \$530 #raskinuti/raskinuh/raskinuo
          - 2.4.2.4.2.1.2. ostali dočetc i
            - 2.4.2.4.2.1.2.1. prad. na -nuo \$531 #ogreznuti/ogreznuo
            - 2.4.2.4.2.1.2.2. prad. i na -Vo

2.4.2.4.2.1.2.2.1. pradž. na -(a)o \$532  
#ogreznuti/ogrezao

2.4.2.4.2.1.2.2.2. pradž. na -eo, -uo \$533  
#uvenuti/uveo, istrunuti/istruo


2.4.2.4.2.2. aorist i na -oh \$ #raskinuti/raskidoh,  
ogreznuti, ogrezoh

2.4.2.4.2.2.1. dočetak osnove na -dne-, -tne-, -bne-, -pne-  
\$534 #raskinuti/raskidoh/raskinuo

2.4.2.4.2.2.2. ostali dočeci \$535 #ogreznuti/ogrezoh

2.4.2.5. dočetak v-e > j-e

2.4.2.5.1. impr. na -o

2.4.2.5.1.1. nesvršeni \$536 

2.4.2.5.1.2. svršeni \$537

2.4.2.5.2. impr. na -i

2.4.2.5.2.1. nesvršeni \$538

2.4.2.5.2.2. svršeni \$539

2.4.2.6. dočetak C-e > Cjot-e, prez. -em

2.4.2.6.1. impr. na -i

2.4.2.6.1.1. nesvršeni \$540

2.4.2.6.1.2. svršeni \$541

2.4.2.6.2. impr. na -o

2.4.2.6.2.1. nesvršeni \$542

2.4.2.6.2.2. svršeni \$543

## 4.4. Komparacija

### 4.4.1. Komparativ

#### 1. Komparativ

- 1.1.osnova + s-j- \$431 (lěp+s-j-i > ljep+š-0-)
- 1.2.osnova + l-j-
  - 1.2.1.nekraćena osnova \$432 (grub+l-j- > grub+lj-0-)
  - 1.2.2.kraćena osnova \$433 (dubok+l-j- > dub+lj-0-)
- 1.3.osnova + 0-j-i
  - 1.3.1.nekraćena osnova \$434, \$435 (ljut+0-j- > ljuć+0-0-)
  - 1.3.2.kraćena osnova \$436, \$437 (visok+0-j- > viš+0-0-)
  - 1.3.3.kraćena osnova, nepostojano a \$438 (kratk+0-j- > krać+0-0-)
  - 1.3.4.nekraćena osnova, nepostojano a \$439, \$440 (běsn+0-j- > bješnj+0-0-)
- 1.4.osnova + i-j-i \$441 (gibk+i-j- > gipk+i-j-)



### 4.4.2. Superlativ

#### 2. Superlativ



Oblici superlativa jednaki su komparativnim oblicima s predmetkom (ili dijelom konfiksa za superlativ) *naj-*.

## 5. RAČUNALNI MODEL GENERIRANJA OBLIKA

Pregled važnijih metoda računalne obrade morfologije prirodnih jezika dâan je iscrpno u Tadić (1992:18-36) stoga ga ovdje nema potrebe ponavljati.

Ovdje prihvaćena metoda za odsječak koji obrađuje uzorke promjene imenica preuzeta je iz istoga izvora.

Leksikonski je, međutim, popis proširen novom građom iz dvaju izvora.

Prvi od njih čini do sada lematizirani dio jednomilijunskoga (1M)

Moguševa korpusa. Drugi izvor čini popis natuknica Anićeva *Rječnika hrvatskoga jezika*. Razlog za odabir ovih dvaju izvora bio je isključivo

praktične prirode — oba su, naime, bila dostupna na magnetskom mediju

tako da te riječi nije trebalo utipkavati. Za buduća istraživanja i

proširivanja leksikonske baze svakako bi valjalo razmotriti uključivanje

građe iz ostalih izvora — višemilijunskoga korpusa hrvatskoga i ostalih

rječnika koje se bilo zagledalo (npr. Deanovićev i Jernejev, Benešićev itd.) i

one koje se još nije stiglo ozbiljno konzultirati (Akademijin).

### 5.1. Definicija i organizacija podataka

Podaci su u računalnoj izvedbi sustava razdijeljeni u tri popisa/propisa:

1. popis nastavaka po uzorcima
2. popis transformacija po uzorcima
3. popis osnova tj. leksikon

Popisi (ujedno i propisi) nastavaka i transformacija konačni su onoliko koliko je sâm model konačan. To, dakako, ne znači da su nepromjenljivi u slučajevima novih potvrda ili protuprimjera. Za razliku od njih, popis osnova otvoren je i koncipiran je tako da se može nadopunjavati novim jedinicama.

### 5.1.1. Nastavci

Popis nastavaka po uzorcima određuje distribuciju nastavaka za svaki oblik svakoga uzorka. Jednoznačno se svaki nastavak imeničnih deklinacijskih uzoraka može definirati kao uređena četvorka:

$$\text{nastavak} := \langle u, pr, b, pa \rangle$$

gdje je  $u$  uzorak (1-404),  $pr$  paradigma (1-4),  $b$  broj (1-2),  $pa$  padež (1-7). Za pridjevske deklinacijske uzorke definicija izgleda ovako:

$$\text{nastavak} := \langle u, pr, r, b, pa \rangle$$

gdje je  $u$  uzorak (451-492),  $pr$  paradigma (1-5),  $r$  rod (1-3),  $b$  broj (1-2),  $pa$  padež (1-7). U konjugacijskim uzorcima nastavci su određeni relacijom:

$$\text{nastavak} := \langle u, v, b, l/r \rangle$$

gdje je  $u$  uzorak (500-653),  $v$  »vrijeme« (1-9),  $b$  broj (1-2),  $l/r$  lice ili rod u slučaju glagolskih pridjeva (1-3). Valja napomenuti da »vrijeme« ne znači gramatičko vrijeme već pokriva organizaciju onih dijelova glagolske paradigme koji pokazuju podstrukturu gdje tri oblika u jednini slijede tri oblika u množini. Takvi su svi finitni oblici s kategorijama broja i lica, ali i oblici glagolskih pridjeva s kategorijama broja i roda. Kad hrvatski ne bi imao tri roda i tri lica (što je poznajući jezične univerzalijske malo vjerojatno) takva izomorfnost ne bi se mogla iskoristiti za sažetiju pohranu podataka kao što je to u ovom slučaju.

Komparativni su nastavci definirani jednostavnom relacijom:

$$\text{nastavak} := \langle u, s \rangle$$

gdje je  $u$  uzorak (431-441), a  $s$  stupanj komparacije (1-3). U praksi se to svodi samo na pohranjivanje nastavka za komparativ jer su pozitiv i komparativ zalihosni.

Podaci definirani prethodnim uređenim četvorkama i petorkama mogu se pohraniti u matricu s četiri (tj. pet) dimenzija, no zbog praktičnih razloga pohranjeni su u dvodimenzionalnu. Horizontalno »čitanje« takve matrice čine uzorci, a vertikalno pojedine paradigme, (rodovi,) brojevi, padeži; odnosno, »vremena«, brojevi, lica/rodovi.

[illegible]



[illegible]





Nastavci konjugacije:

[illegible]

[illegible]

### 5.1.2. Preoblike

#### 5.1.2.1. Temeljna struktura preoblika

Preoblike se u sustavu GENIMEN, kao što je to već bilo rečeno, primjenjuju isključivo na osnove. Za svaki je oblik svakoga uzorka svake vrste promjene određeno kako se osnova mora preobličiti ne bi li spojena s nastavkom mogla tvoriti riječ na izlaznoj ili preobličenoj razini. Da bi se od temeljnoga lika osnove dobio preobličeni, najčešće nije dovoljna samo jedna nego više uzastopnih preoblika. Takav niz preoblika sastoji se od jedne do najviše četiri jednostavne preoblike.

Popis preoblika po uzorcima određuje distribuciju takvih nizova preoblika za svaki oblik svakoga uzorka. Jednoznačno se svaki niz preoblika u imeničnih deklinacijskih uzoraka može definirati kao uređena četvorka:

$$\text{preoblike} := \langle u, pr, b, pa \rangle$$

gdje je *u* uzorak (1-404), *pr* paradigma (1-4), *b* broj (1-2), *pa* padež (1-7). Za pridjevske deklinacijske uzorke definicija niza transformacija za svaki oblik izgleda ovako:

$$\text{preoblike} := \langle u, pr, r, b, pa \rangle$$

gdje je *u* uzorak (451-492), *pr* paradigma (1-5), *r* rod (1-3), *b* broj (1-2), *pa* padež (1-7). U konjugacijskim uzorcima nizovi preoblika određeni su relacijom:

$$\text{preoblike} := \langle u, v, b, l/r \rangle$$

gdje je *u* uzorak (500-653), *v* »vrijeme« (1-9), *b* broj (1-2), *l/r* lice ili rod u slučaju glagolskih pridjeva (1-3). Komparativni su nizovi transformacija osnove definirani jednostavnom relacijom:

$$\text{preoblike} := \langle u, s \rangle$$

gdje je  $u$  uzorak (431-441), a  $s$  stupanj komparacije (1-3). U praksi se to svodi samo na pohranjivanje nizova transformacija za komparativ jer su pozitiv i komparativ zalihosni. Podaci definirani prethodnim uređenim četvorkama i petorkama pohranjeni su na isti način kao i nastavci iz popisa nastavaka. Ono što se može uočiti već na prvi pogled jest potpuna izomorfnost popisa nastavaka i popisa transformacija osnove.

Time što je red preoblika u nizu određen, određen je i red njihove primjene na osnovu. Same su preobliske određene kao operacije nad pismenima koja čine sastav osnove. Te se operacije sastoje od:

1. izbacivanja pojedinih pismenâ
2. zamjena pojedinih pismenâ ili grupâ pismenâ pojedinim pismenima ili grupama pismena
3. dodavanje pismena ili grupa pismena na kraj ili prije kraja osnove

Transformacije su podijeljene u dvije grupe: one koje se izvršuju bezuvjetno tj. bez provjere okoline na koju se moraju primijeniti (te su preobliske u popisu obilježene zvjezdicom) i one koje se izvršuju ako je zadovoljen njihov kontekst — fakultativne transformacije. Ne treba smetnuti s uma da je opći kontekst primjene preoblika određen samim uzorkom jer se neka transformacija ne može ni pojaviti nekontrolirano, mimo točno određenoga oblika u uzorku. Time su se stekli uvjeti za relativno jednostavne, gotovo bezokolinske preobliske koje djeluju automatski.

Samih je preoblika u različitim vrstama promjene različit broj (18 se rabi u imeničnoj deklinaciji, 24 u konjugaciji itd.) Stoga se ovdje daje kumulativni (p)opis pojedinačnih transformacija. One su jednoznačno kodirane brojevima ili slovima i popraćene mnemotehničkim imenom:

- 1.\* UmetniA
2. Sibilarizacija

3. Palatalizacija
4. Bipalatalizacija
5. Jotacija
6. VokaliziranjeL
- 7.\* OdbaciKonacniC
8. IzbaciUnutarnjiC
- 9.\* ProsiriSJ
- a. Obezvucavanje
- b. KratiJatUI
- d.\* ProsiriSEr
- c. JatuU ili OsnovauCovjek
- e. StegniV
- f. StegniC
- g. ObezvucavanjeBezOkoline
- h.\* UmetniA2
- i. SrediJat
- j. EuO
- k. PrijevojPuninaE
- l. PrijevojPuninaO
- m.\* OdbaciPretkonacniC
- n. EuA
- o. EuI
- p. EuU
- q. JatuI
- r. JatuE
- s. IuE
- t. OuA
- u. VEuUjE
- v.\* UmetniV
- w.\* IzbaciPretpretkonacniC
- x. VuJ
- y.\* UmetniUv
- z.\* DodajVe

Dio je preoblika s popisa već iznesen u GRAM gdje se daje popis preoblika koje sudjeluju u izvođenju oblika imenične deklinacije, a to su: obezvučena osnova, okrnjena osnova, vokalizirana osnova,<sup>117</sup> palatalizirana osnova, bi-

---

<sup>117</sup> cf. GRAM, str. 498, t. 124.

palatalizirana osnova,<sup>118</sup> sibilizirana osnova, okrnjena sibilizirana osnova,<sup>119</sup> osnova proširena s *-j-*,<sup>120</sup> osnova proširena s *-er-*,<sup>121</sup> a negdje se govori samo o glasovnim promjenama na granici morfema.<sup>122</sup> No te navedene alternacije nisu dovoljne da se opišu sve transformacije ni u onom dijelu računalnoga modela koji obrađuje uzorke imenične deklinacije, stoga slijede opisi transformacija. Poredak je transformacija onaj kojim se u izgrađivanju modela pojavljivala potreba za njihovom primjenom.

#### 5.1.2.2. Opis preoblika

**UmetniA** transformacija umeće nepostojano *-a-* prije zadnjega pismena osnove. To se može zapisati kao:

$$\dots p_{n-1} p_n > \dots p_{n-2} a p_n$$

gdje je  $p_n$  zadnje pisme u osnovi, a  $p_{n-1}$  predzadnje, a ... označuje broj pismena veći ili jednak 0. Pravilo se primjenjuje bezuvjetno. Da nema zahtjeva za uporabom standardnoga alfabeta<sup>123</sup> mogao bi se zapisati konstrukt nepostojanoga *-a-* u osnovi što bi, međutim, vodilo većem broju primjena transformacije brisanja nego u obratnome slučaju. Tako postoji transformacija umetanja *-a-* u manjem broju padeža. Prigovor da je umetanje nepostojanoga *-a-* *creatio ex nihilo* ne stoji jer je ta transformacija, kao i sve ostale, striktno vezana uz deklinacijski uzorak kojim je

---

<sup>118</sup> cf. GRAM str. 501, t. 135. i str. 502, t. 136a.

<sup>119</sup> cf. GRAM, str. 508, t. 148.

<sup>120</sup> cf. GRAM, str. 520, t. 202.

<sup>121</sup> cf. GRAM, str. 569, t. 274.

<sup>122</sup> cf. GRAM, str. 604, t. 374, t. 375, t. 376, t. 377.

<sup>123</sup> v. 2.2.5. gore.

zapravo opisan kontekst. Uopće je težnja u stvaranju modela bila da se broj potrebnih transformacija svede na minimum zbog što kraćega puta i vremena preobličavanja morfotaktičkoga niza u preoblićeni niz.

**Sibilarizacija** mijenja *-k, -g, -h* na kraju osnove u *-c, -z, -s*:

$$\dots p_n > \dots P_n / p = [g, h, k], P = [z, s, c]$$

gdje je  $p_n$  zadnje pisme u osnovi,  $P_n$  preoblićeno zadnje pisme u osnovi, a moguće vrijednosti  $p$  i  $P$  određene su korespondirajućim nizovima unutar [] nakon oznake za opis konteksta /.

**Palatalizacija** mijenja *-k, -g, -h, -c, -z* na kraju osnove u *-č, -ž, -š, -ć, -ž*:

$$\dots p_n > \dots P_n / p = [c, g, h, k, z], P = [\check{c}, \check{z}, \check{s}, \check{c}, \check{z}]$$

**Bipalatalizacija** na kraju osnove mijenja *-stk, -zdk, u -šč; -sl, -zl, -sn, -zn u -šl, -žl, -šn, -žn; -sć, -zć, -žć, -sč, -zč, -žč u -šč; čt- u št-*:

$$\dots p_{n-2} p_{n-1} p_n > \dots P_{n-1} P_n / p_{n-2}+p_{n-1}+p_n = [stk, zdk], \\ P_{n-1}+P_n = [\check{s}\check{c}]$$

$$\dots p_{n-1} p_n > \dots P_{n-1} P_n / p_{n-1}+p_n = [sl, zl, sn, zn, s\acute{c}, z\acute{c}, \check{z}\acute{c}, \\ s\check{c}, z\check{c}, \check{z}\check{c}, \check{c}t], P_{n-1}+P_n = [\check{s}l, \check{z}l, \\ \check{s}n, \check{z}n, \check{s}\acute{c}, \check{s}\acute{c}, \check{s}\check{c}, \check{s}\check{c}, \check{s}\check{c}, \check{s}t]$$

gdje se + koristi za kontekste sastavljene od više pismena. U cijeloj se klasifikaciji *-st* i *-zd* (kao i *-št* i *-žd*) ponašaju poput jednoga suglasnika a ne suglasničke grupe stoga se ne treba čuditi tako širokom kontekstu u prvom zapisu bipalatalizacije. On se gledajući ponašanje *-st* i *-zd* svodi na drugi zapis.<sup>124</sup>

---

<sup>124</sup> v. gore 2.2.5.1. i detaljnije o toj problematici u Silić (1992).

**Jotacija** dodaje *lj* iza *b, p, v, f, m*; *j* iza *l, n*; mijenja *s, h, g, z, t, d, c, h* u konačnom ili pretkonačnom pismenu osnove u *š, š, ž, ž, ć, đ, č, č*:

$$\begin{aligned} \dots p_n \dots &> \dots p_n \dots \quad l j / \quad p_n = [b, p, v, f, m] \\ \dots p_n \dots &> \dots p_n \dots \quad j / \quad p_n = [l, n] \\ \dots p_n \dots &> \dots P_n \dots / \quad p_n = [s, h, g, z, t, d, c, h], \\ & \quad P_n = [\check{s}, \check{s}, \check{z}, \check{z}, \acute{c}, \check{d}, \check{c}, \check{c}] \end{aligned}$$

gdje su  $p_n$  i  $P_n$  konačno ili pretkonačno pisme osnove.

**Vokaliziranje** *L* pretvara -*l*- na predzadnjem ili -*l* na zadnjem mjestu osnove u -*o*- ili -*o*:

$$\dots p_{n-1} \ p_n > \dots P_{n-1} \ p_n / p_{n-1} = [1], \ P_{n-1} = [0]$$

$$\dots p_n > \dots P_n / p = [1], \ P = [0]$$

**OdbaciKonacniC** odbacuje od osnove njezino dočetno pisme:

$$\dots p_{n-1} \ p_n > \dots p_{n-1}$$

**IzbaciUnutarnjiC** na kraju osnove mijenja -*stk*, -*zdk* u -*sk*; -*štk*, -*ždk* u -*šk*; -*tc*, -*dc* u -*c*; -*stl* u -*sl*:

$$\begin{aligned} \dots p_{n-2} \ p_{n-1} \ p_n &> \dots p_{n-1} \ p_n / \quad p_{n-2}+p_{n-1}+p_n = [stk, zdk, štk, \\ &\quad ždk, stl] , \quad p_{n-1}+p_n = [sk, sk, \\ &\quad šk, šk, sl] \\ \dots p_{n-1} \ p_n &> \dots p_n / \quad p_{n-1}+p_n = [dc, tc] , \quad p = [c, c] \end{aligned}$$

**ProsiriSJ** osnovi dodaje -j:

$$\dots p_n > \dots p_n^j$$



**Obezvucavanje** pretvara  $C_1$  u njegov bezvučni ekvivalent u prvom suglasničkom paru  $C_1C_2$  zdesna:

$$\begin{aligned} \dots p_{n-1} p_n \dots &> \dots P_{n-1} p_n \dots / p_{n-1} = [b, d, dž, đ, g, z, ž], \\ p_n &= [c, č, ć, h, k, p, s, š, t], \\ p_{n-1} &= [p, t, č, ć, k, s, š] \end{aligned}$$

**KratiJatUI** krati -ije-, -je- u -i-:

$$\begin{aligned} \dots p_{n-2} p_{n-1} p_n \dots &> \dots P_n \dots / p_{n-2}+p_{n-1}+p_n = [ije], P_n = [i] \\ \dots p_{n-1} p_n \dots &> \dots P_n \dots / p_{n-1}+p_n = [je], P_n = [i] \end{aligned}$$

**OsnovaUCovjek** čitavu osnovu pretvara u osnovu *čovjek*-:

$$p_1 \dots p_n > P_1 P_2 P_3 P_4 P_5 / P_1+P_2+P_3+P_4+P_5 = [\text{čovjek}]$$

**ProsiriSEr** osnovi dodaje -er-:

$$\dots p_n > \dots p_n e r$$

**StegniV** skraćuje u jedan dva ista uzastopna samoglasnika na kraju i pred krajem osnove:

$$\begin{aligned} \dots p_{n-1} p_n &> \dots P_n / p_{n-1} = p_n = [a, e, i, o, u], P_n = [a, e, i, o, u] \\ \dots p_{n-2} p_{n-1} p_n &> \dots P_{n-1} p_n / p_{n-2} = p_{n-1} = [a, e, i, o, u], \\ P_{n-1} &= [a, e, i, o, u] \end{aligned}$$

**StegniC** skraćuje u jedan dva ista uzastopna suglasnika na kraju i pred krajem osnove:

$$\begin{aligned} \dots p_{n-1} p_n &> \dots P_n / p_{n-1} = p_n = [b, d, dž, đ, g, z, ž, c, č, ć, h, k, p, \\ &\quad s, š, t], P_n = [b, d, dž, đ, g, z, ž, c, č, ć, h, k, \\ &\quad p, s, š, t] \\ \dots p_{n-2} p_{n-1} p_n &> \dots P_{n-1} p_n / p_{n-2} = p_{n-1} = [b, d, dž, đ, g, z, ž, \\ &\quad c, č, ć, h, k, p, s, š, t], \end{aligned}$$

$$P_{n-1} = [b, d, dž, đ, g, z, ž, c, č, \\ \acute{c}, h, k, p, s, š, t]$$

**ObezvucavanjeBezOkoline** pretvara C u njegov bezvučni ekvivalent bez provjere okoline:

$$\dots P_n \dots > \dots P_n \dots / \quad \begin{aligned} P_n &= [b, d, dž, đ, g, z, ž], \\ P_n &= [p, t, č, \acute{c}, k, s, š] \end{aligned}$$

**UmetniA2** umeće nepostojano -a- ispred predzadnjeg pismena u osnovi (pokriva jedinstven slučaj imenice *zajturak/zajutarka*):

$$\dots P_{n-2} P_{n-1} P_n > \dots P_{n-2} a P_{n-1} P_n$$

**SrediJat** umjesto ě ostvaruje se e ili je ovisno o kontekstu:

$$\begin{aligned} \dots P_{n-2} P_{n-1} P_n \dots > \dots P_{n-2} P_{n-1} P_n \dots / \quad & P_n = [\acute{e}], \quad P_{n-1} = [r], \\ & P_{n-2} = [b, d, dž, đ, g, z, ž, \\ & c, \acute{c}, \acute{c}, h, k, p, s, š, t, f, j, l, \\ & m, n, r, v, lj, nj], \quad P_n = [e] \\ \dots P_n \dots > \dots P_{n-1} P_n \dots / & P_n = [\acute{e}], \quad P_{n-1} + P_n = [je] \end{aligned}$$

**EuO** zamjenjuje -e na kraju osnove s -o:

$$\dots p > \dots P / p = [e], \quad P = [o]$$

**PrijevojPuninaE** umeće -e- ispred konačnoga pismena osnove:

$$\dots P_{n-1} P_n > \dots P_{n-2} e P_n$$

**PrijevojPuninaO** umeće -o- ispred konačnoga pismena osnove:

$$\dots P_{n-1} P_n > \dots P_{n-2} o P_n$$

**OdbaciPretkonacniC** iz osnove izbacuje pretkonačno pisme:

$$\dots P_{n-1} P_n > \dots P_n$$

***EuA*** zamjenjuje *-e* na kraju osnove s *-a*:

$$\dots p > \dots P / p = [e] , P = [a]$$

***EuI*** zamjenjuje *-e* na kraju osnove s *-i*:

$$\dots p > \dots P / p = [e] , P = [i]$$

***EuU*** zamjenjuje *-e* na kraju osnove s *-u*:

$$\dots p > \dots P / p = [e] , P = [u]$$

***JatuI*** zamjenjuje *-ě* kao jedno od tri konačna pismena u osnovi s *-i*:

$$\dots p > \dots P / p = [ě] , P = [i]$$

$$\dots p_{n-1} p_n > \dots P_{n-1} p_n / p_{n-1} = [ě] , P_{n-1} = [i]$$

$$\dots p_{n-2} p_{n-1} p_n > \dots P_{n-2} p_{n-1} p_n / p_{n-2} = [ě] , P_{n-2} = [i]$$

***JatuE*** zamjenjuje *-ě* kao jedno od tri konačna pismena u osnovi s *-e*:

$$\dots p > \dots P / p = [ě] , P = [e]$$

$$\dots p_{n-1} p_n > \dots P_{n-1} p_n / p_{n-1} = [ě] , P_{n-1} = [e]$$

$$\dots p_{n-2} p_{n-1} p_n > \dots P_{n-2} p_{n-1} p_n / p_{n-2} = [ě] , P_{n-2} = [e]$$

***JatuU*** zamjenjuje *-ě* kao jedno od tri konačna pismena u osnovi s *-u*:

$$\dots p > \dots P / p = [ě] , P = [u]$$

$$\dots p_{n-1} p_n > \dots P_{n-1} p_n / p_{n-1} = [ě] , P_{n-1} = [u]$$

$$\dots p_{n-2} p_{n-1} p_n > \dots P_{n-2} p_{n-1} p_n / p_{n-2} = [ě] , P_{n-2} = [u]$$

***IuE*** zamjenjuje *-i* na kraju osnove s *-e*:

$$\dots p > \dots P / p = [i] , P = [e]$$

**OuA** zamjenjuje -o na pretpretkonačnom mjestu osnove s -a:

$$\dots p_{n-2} p_{n-1} p_n > \dots p_{n-2} p_{n-1} p_n / p_{n-2} = [o], p_{n-2} = [a]$$

**VEuUjE** pretvara -ee, -ie, -oe na kraju osnove u -uje:

$$\dots p_{n-1} p_n > \dots p_{n-2} p_{n-1} p_n / \quad p_{n-1} = [e, i, o], p_n = [e], \\ p_{n-2} + p_{n-1} = [uj]$$

**UmetniV** umeće -v- ispred konačnoga pismena osnove:

$$\dots p_{n-1} p_n > \dots p_{n-2} v p_n$$

**IzbaciPretpretkonacniC** iz osnove izbacuje pretpretkonačno pisme:

$$\dots p_{n-2} p_{n-1} p_n > \dots p_{n-1} p_n$$

**VuJ** zamjenjuje pretkonačno -v- s -j-:

$$\dots p_{n-1} p_n > \dots p_{n-1} p_n / \quad p_{n-1} = [v], p_{n-1} = [j]$$

**UmetniUv** umeće -uv- prije konačnoga pismena osnove:

$$\dots p_{n-1} p_n > \dots p_{n-3} uv p_n$$

**DodajVe** proširuje osnovu s -ve:

$$\dots p_n > \dots p_{n-2} ve$$



P2										P3										P4										P5										P6										P7										P8										P9										P10										P11										P12										P13										P14										P15										P16										P17										P18										P19										P20										P21										P22										P23										P24										P25										P26										P27										P28										P29										P30										P31										P32										P33										P34										P35										P36										P37										P38										P39										P40										P41										P42										P43										P44										P45										P46										P47										P48										P49										P50										P51										P52										P53										P54										P55										P56										P57										P58										P59										P60										P61										P62										P63										P64										P65										P66										P67										P68										P69										P70										P71										P72										P73										P74										P75										P76										P77										P78										P79										P80										P81										P82										P83										P84										P85										P86										P87										P88										P89										P90										P91										P92										P93										P94										P95										P96										P97										P98										P99										P100										P101										P102										P103										P104										P105										P106										P107										P108										P109										P110										P111										P112										P113										P114										P115										P116										P117										P118										P119										P120										P121										P122										P123										P124										P125										P126										P127										P128										P129										P130										P131										P132										P133										P134										P135										P136										P137										P138										P139										P140										P141										P142										P143										P144										P145										P146										P147										P148										P149										P150										P151										P152										P153										P154										P155										P156										P157										P158										P159										P160										P161										P162										P163										P164										P165										P166										P167										P168										P169										P170										P171										P172										P173										P174										P175										P176										P177										P178										P179										P180										P181										P182										P183										P184										P185										P186										P187										P188										P189										P190										P191										P192										P193										P194										P195										P196										P197										P198										P199										P200										P201										P202										P203										P204										P205										P206										P207										P208										P209										P210										P211										P212										P213										P214										P215										P216										P217										P218										P219										P220										P221										P222										P223										P224										P225										P226										P227										P228										P229										P230										P231										P232										P233										P234										P235										P236										P237										P238										P239										P240										P241										P242										P243										P244										P245										P246										P247										P248										P249										P250										P251										P252										P253										P254										P255										P256										P257										P258										P259										P260										P261										P262										P263										P264										P265										P266										P267										P268										P269										P270										P271										P272										P273										P274										P275										P276										P277										P278										P279										P280										P281										P282										P283										P284										P285										P286										P287										P288										P289										P290										P291										P292										P293										P294										P295										P296										P297										P298										P299										P300										P301										P302										P303										P304										P305										P306										P307										P308										P309										P310										P311										P312										P313										P314										P315										P316										P317										P318										P319										P320										P321										P322										P323										P324										P325										P326										P327										P328										P329										P330										P331										P332										P333										P334										P335										P336										P337										P338										P339										P340										P341										P342										P343										P344										P345										P346										P347										P348										P349										P350										P351										P352										P353										P354										P355										P356										P357										P358										P359										P360										P361										P362										P363										P364										P365										P366										P367										P368										P369										P370										P371										P372										P373										P374										P375										P376										P377										P378										P379										P380										P381										P382										P383										P384										P385										P386										P387										P388										P389										P390										P391										P392										P393										P394										P395										P396										P397										P398										P399										P400										P401										P402										P403										P404										P405										P406										P407										P408										P409										P410										P411										P412										P413										P414										P415										P416										P417										P418										P419										P420										P421										P422										P423										P424										P425										P426										P427										P428										P429										P430										P431										P432										P433										P434										P435										P436										P437										P438										P439										P440										P441										P442										P443										P444										P445										P446										P447										P448										P449										P450										P451										P452										P453										P454										P455										P456										P457										P458										P459										P460										P461										P462										P463										P464										P465										P466										P467										P468										P469										P470										P471										P472										P473										P474										P475										P476										P477										P478										P479										P480										P481										P482										P483										P484										P485										P486										P487										P488										P489										P490										P491										P492										P493										P494										P495										P496										P497										P498										P499										P500										P501										P502										P503										P504										P505										P506										P507										P508										P509										P510										P511										P512										P513										P514										P515										P516										P517										P518										P519										P520										P521										P522										P523										P524										P525										P526										P527										P528										P529										P530										P531										P532										P533										P534										P535										P536										P537										P538										P539										P540										P541										P542										P543										P544										P545										P546										P547										P548										P549										P550										P551										P552										P553										P554										P555										P556										P557										P558										P559										P560										P561										P562										P563										P564										P565										P566										P567										P568										P569										P570										P571										P572										P573										P574										P575										P576										P577										P578										P579										P580										P581										P582										P583										P584										P585										P586										P587										P588										P589										P590										P591										P592										P593										P594										P595										P596										P597										P598										P599										P600										P601										P602										P603										P604										P605										P606										P607										P608										P609										P610										P611										P612										P613										P614										P615										P616										P617										P618										P619										P620										P621										P622										P623										P624										P625										P626										P627										P628										P629										P630										P631										P632										P633										P634										P635										P636										P637										P638										P639										P640										P641										P642										P643										P644										P645										P646										P647										P648										P649										P650										P651										P652										P653										P654										P655										P656										P657										P658										P659										P660										P661										P662										P663										P664										P665										P666										P667										P668										P669										P670										P671										P672										P673										P674										P675										P676										P677										P678										P679										P680										P681										P682										P683										P684										P685										P686										P687										P688										P689										P690										P691										P692										P693										P694										P695										P696										P697										P698										P699										P700										P701										P702										P703										P704										P705										P706										P707										P708										P709										P710										P711										P712										P713										P714										P715										P716										P717										P718										P719										P720										P721										P722										P723										P724										P725										P726										P727										P728										P729										P730										P731										P732										P733										P734										P735										P736										P737										P738										P739										P740										P741										P742										P743										P744										P745										P746										P747										P748										P749										P750										P751										P752										P753										P754										P755										P756										P757										P758										P759										P760										P761										P762										P763										P764										P765										P766										P767										P768										P769										P770										P771										P772										P773										P774										P775										P776										P777										P778										P779										P780										P781										P782										P783										P784										P785										P786										P787										P788										P789										P790										P791										P792										P793										P794										P795										P796										P797										P798										P799										P800										P801										P802										P803										P804										P805										P806										P807										P808										P809										P810										P811										P812										P813										P814										P815										P816										P817										P818										P819										P820										P821										P822										P823										P824										P825										P826										P827										P828										P829										P830										P831										P832										P833										P834										P835										P836										P837										P838										P839										P840										P841										P842										P843										P844										P845										P846										P847										P848										P849										P850										P851										P852										P853										P854										P855										P856										P857										P858										P859										P860										P861										P862										P863										P864										P865										P866										P867										P868										P869										P870										P871										P872										P873										P874										P875										P876										P877										P878										P879										P880										P881										P882										P883										P884										P885										P886										P887										P888										P889										P890										P891										P892										P893										P894										P895										P896										P897										P898										P899										P900										P901										P902										P903										P904										P905										P906										P907										P908										P909										P910										P911										P912										P913										P914										P915										P916										P917										P918										P919										P920										P921										P922										P923										P924										P925										P926										P927										P928										P929										P930										P931										P932										P933										P934										P935										P936										P937										P938										P939										P940										P941										P942										P943										P944										P945										P946										P947										P948										P949										P950										P951										P952										P953										P954										P955										P956										P957										P958										P959										P960										P961																			
----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	-----	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

[illegible]







[illegible]

### 5.1.3. Leksikon

#### 5.1.3.1. Uloga leksikona u modelu

Leksičke jedinice čine najveći dio govornikova znanja jezika, pa ipak nije uvijek potpuno jasno koje se jedinice i u kakvu obliku u njemu moraju pojaviti. Kad bi se u ovome radu polazilo od toga da se opiše jezično znanje govornika i to ona njegova razina koja uključuje sposobnost proizvodnja različitih oblika riječi, onda bi se vrlo brzo našli između tri više ili manje suprotstavljena mišljenja:<sup>125</sup>

1. svi su oblici riječi pohranjeni u leksikonu (izravno);
2. niti jedan oblik riječi nije pohranjen u leksikonu već se oni izvode s pomoću pravila neovisnih o leksikonskome ulazu koja uzimlju bazičnu formu kao ulaz, a konačni oblik kao izlaz;
3. kombinacija 1. i 2. mišljenja koja dopušta pohranu nekih oblika (najčešće »nepravilnih«) u leksikon, ali i neovisna pravila za proizvodnju pravilnih oblika.

Psiholingvistička su istraživanja dokazala da doista postoji pohrana nekih, i to upravo »nepravilnih« oblika, izravno u leksikon, ali uz istodobnu svijest o njihovoj pripadnosti paradigmi.<sup>126</sup>

Navodnici oko termina *nepravilni* u oba su gornja navođenja naknadno ovdje umetnuti. Time se želi pokazati da ako se prihvati paradigmatski oblik organizacije leksikona (kao u Aronoff (1976) koji upravo postojanjem supletivnih oblika i njihovim smještanjem u paradigmatske odnose dokazuje postojanje na razini fleksije uređenoga skupa odnosa tj.

---

<sup>125</sup> Stemberger – MacWhinney (1988:101).

<sup>126</sup> Stemberger – MacWhinney (1988:102)

paradigme, v. **3.** gore), a što ovaj rad zastupa od početka, izravno se dolazi u kontradikciju s prvim stavom koji u potpunosti isključuje morfologiju. Kad bi se prvi stav prihvatio, teško bi se mogli prepoznati oblici riječi iz Carrollove pjesmice citirani u **3.1.** gore. Takvih riječi ne bi bilo u leksikonu. Drugi je stav teško branjiv zbog psiholingvističkih potvrda, tako da se najupotrebljivijim za opis stvarnoga jezičnog znanja govornika čini treći stav.

Na sreću, cilj ovoga rada nije opis stvarnoga jezičnog znanja govornika koji je kadar proizvoditi oblike riječi, već je cilj takav opis morfologije koji omogućuje njegovo pretakanje u računalni model kadar proizvoditi oblike riječi. Stoga ovdje psiholingvističke činjenice nisu relevantne već je relevantna informatički što učinkovitija organizacija modela koji, kao i svi ostali sustavi za obradu morfologije prirodnih jezika, mora osim pravila kombiniranja jedinica (propisa) imati i leksikon (popis jedinica).

U naravi je leksikona da bude popisom, iscrpnim ako je to moguće. Opsezi su takvih popisa znatni i stoga je potrebno u što kraćem obliku smjestiti u njega što je moguće više podataka. To je prvi zahtjev koji je odlučivao o organizaciji podataka u leksikonu. Drugi je zahtjev bio »prirodnost« leksikonskih ulaza tj. natuknica i zapisa osnova. Ta je prirodnost ostvarena prije svega uporabom standardnoga pravopisa i time je izbjegnuto zapisivanje posebnim alfabetom natuknica u obliku apstraktnih korijena/osnova što je prisutno u nekim sustavima za obradu morfologije.<sup>127</sup>

Za razliku od mnogih prijedloga za računalnu organizaciju leksikona koja bi bila dovoljno fleksibilna kako bi omogućila njegovu višesvrhovitu uporabu (Byrd i drugi (1987), Dumitrescu (1992), Domenig (1990), Molbæk Hansen (1989), Russel i drugi (1986), Boguraev – Briscoe (1989)), ovdje je leksikon organiziran u najjednostavnijem mogućem obliku. Takvo

---

<sup>127</sup> cf. Jäppinen-Ylilampi (1986), str. 271.

je ograničenje rezultat namjerne »nedorečenosti« računalne izvedbe u smislu visokoučinkovitog organiziranja podataka ili optimiziranja programa. Željelo se računalni model zadržati što transparentnijim u odnosu na njegovu lingvističku osnovicu, koja bi se providnost zacijelo izgubila da su uključeni informatički postupci optimiziranja podataka (kompresija npr.). Konačno, ovaj leksikona i ne postoji zato da bi bio višesvrhovitim pomagalom, nego zato da bi mogao pružiti ulazne podatke za sustav za generiranje oblika riječi u hrvatskome — GENOBLIK.

#### 5.1.3.2. Struktura leksikonskoga ulaza

Jedan leksikonski ulaz čini:

1. natuknica
2. osnova
3. broj uzorka promjene
4. \*rod u imenica (gdje se on ne može predvidjeti iz uzorka) ili oznaka neprijelaznosti u glagola
5. \*broj u imenica (1 obilježuje da je imenica *singulare tantum*, a 2 da je *plurale tantum*)
6. \*značenje (ukoliko je potrebno za razlikovanje dviju natuknica)

Zvjezdicom su označene fakultativne kategorije podataka. Također je moguće natuknicu obilježiti indeksom zbog potrebe razlikovanja istopisnica (v. bar1 'mjerna jedinica za tlak' i bar2 'lokal'). Broj uzorka promjene razložen je na brojeve uzoraka triju osnovnih vrsta promjene te ima opći oblik:

nnn \*(, nnn)/ nnn \*(, nnn)/ nnn \*(, nnn)

gdje je \*(, nnn) fakultativni broj dodatnih uzoraka po kojima se mogu izvoditi oblici te natuknice. Maksimalni je dopušteni broj različitih uzoraka za svaku vrstu, postavljen zbog tehničkih razloga, četiri. Slijedi izvadak iz leksikona kao primjer:

mještatin mještatin 3/0/0	vijek1 včk 43/0/0
Turčin Turk 4/0/0	vijek2 vijek 43/0/0
Ugrin Ugr 4/0/0	vlak vlak 43/0/0
crv crv 5/0/0	čvarak čvark 50/0/0
mrav mrav 5/0/0	denjak denjk 51/0/0
gost gost 6/0/0	lijevak lčvk 51/0/0
čovjek ljud 7/0/0	prašak prašk 51/0/0
život život 8/0/0	pupak pupk 51/0/0
atom atom 8/0/0	trošak trošk 51/0/0
automobil automobil 8/0/0	višak višk 51/0/0
bar1 bar 8/0/0 m jedinica	vršak vršk 51/0/0
bar2 bar 9/0/0 m lokal	zamak zamk 51/0/0
dan dan 8/0/0	zglavak zglavk 51/0/0
dječak dječak 40/0/0	čošak čošk 51/0/0
morf morf 33/0/0	čunak čunk 51/0/0
pertl pertl 33/0/0	sinak sink 52/0/0
pinkl pinkl 33/0/0	zajuttrak zajutrk 53/0/0
pinzl pinzl 33/0/0	đavolak đavolk 54/0/0
tromb tromb 33/0/0	posalak posalk 56/0/0
verb verb 33/0/0	stalak stalk 56/0/0
volt volt 33/0/0	volak volk 58/0/0
bik bik 41/0/0	zaselak zaselk 60/0/0
rak1 rak 41/0/0 m životinja	cucak cuck 62/0/0
rak2 rak 43/0/0 m bolest	natucak natuck 64/0/0
vuk vuk 41/0/0	vražićak vražićk 66/0/0
bezobrazluk bezobrazluk 42/0/0	hrčak hrčk 67/0/0
jezik jezik 42/0/0	mačak mačk 67/0/0
korak korak 42/0/0	smrčak smrčk 67/0/0
lješnik lješnik 42/0/0	svračak svračk 67/0/0
lješnjak lješnjak 42/0/0	maslačak maslačk 68/0/0
oblak oblak 42/0/0	stećak stećk 68/0/0
oblik oblik 42/0/0	čičak čičk 68/0/0
obluk obluk 42/0/0	ručak ručk 69/0/0
rizik rizik 42/0/0	točak točk 69/0/0
zapisnik zapisnik 42/0/0	tučak tučk 69/0/0
brak brak 43/0/0	gusak gusk 71/0/0
brk brk 43/0/0	naglasak naglask 72/0/0
buk buk 43/0/0	blijesak blčsk 73/0/0
krak krak 43/0/0	pisak pisk 73/0/0
krik krik 43/0/0	pljusak pljusk 73/0/0
luk luk 43/0/0	prasak prask 73/0/0
mak mak 43/0/0	spisak spisk 73/0/0
mrak mrak 43/0/0	trijesak trčsk 73/0/0
oluk oluk 43/0/0	visak1 visk 73/0/0 m sprava
puk puk 43/0/0	visak2 visk 73/0/0 m rzaaj
rijek rčk 43/0/0	vosak vosk 73/0/0

vrijesak vrěsk	73/0/0	manioka maniok	345/0/0
golubak golubk	86/0/0	žbuka žbuk	346/0/0
držak držk	88/0/0	vlast vlast	388/0/0
medvjedak medvjedk	90/0/0	vrijednost vrijednost	388/0/0
dohodak dohodk	92/0/0	časť časť	388/0/0
pregradak pregradk	92/0/0	kost kost	389/0/0
redak redk	92/0/0	ravan ravn	390/0/0
zarudak zarudk	96/0/0	plijesan plijesn	391/0/0
grozdak grozdk	100/0/0	bojazan bojazn	392/0/0
zviždak zviždk	100/0/0	pogibao pogibl	393/0/0
omedak omedk	104/0/0	misao misl	394/0/0
orozak orozk	106/0/0	pomrzao pomrzi	395/0/0
svezak svezk	108/0/0	izrastao izrastl	396/0/0
drozak drozg	110/0/0	kći kć	397/0/0
brizak brizg	111/0/0	Hrvatska Hrvatsk	402/0/0 f1
mozak mozg	111/0/0	draga2 drag	402/0/0 f mila
katafalk katafalk	114/0/0	mlada mlad	402/0/0
obelisk obelisk	114/0/0	slatkol slatk	403/0/0 n džem
štrajk štrajk	115/0/0	drag drag	451,452/0/434
Bask Bask	116/0/0	drago drag	0/0/434
jastog jastog	120/0/0	brz brz	451,452/0/434
nosorog nosorog	120/0/0	brzo brz	0/0/434
beg beg	121/0/0	dug2 dug	451,452/0/432,434 . dugačak
bog bog	121/0/0	dugo dug	0/0/432,434
drug drug	121/0/0	lijep lěp	451,452/0/431
plačidrug plačidrug	121/0/0	lijepo lěp	0/0/431
polubog polubog	121/0/0	lak lak	451,452/0/431 . lagan
vrag vrág	121/0/0	lako lak	0/0/431
bubreg bubreg	122/0/0	lagan lagan	451,452/0/441
razlog razlog	122/0/0	lagano lagan	0/0/441
ulog ulog	122/0/0	mek mek	451,452/0/431
brijeg brěg	123/0/0	meko mek	0/0/431
dugl dug	123/0/0	malen malen	451,452/0/434
rog rog	123/0/0	maleno malen	0/0/434
snijeg sněg	123/0/0	malo malen	0/0/434
udio udjel	265/0/0	velik velik	451,452/0/434
udjel udjel	265/0/0	veliko velik	0/0/434
đavo đavol	266/0/0	širok širok	451,452/0/436
đavol đavol	266/0/0	široko2 širok	0/0/436
prijesto prijestol	268/0/0	dobar dobr	453,452/0/435
prijestol prijestol	268/0/0	dobro2 dobr	0/0/435
đavao đavl	270/0/0	ljubak ljubk	454,455/0/441
djetao djetl	271/0/0	ljupko ljubk	0/0/441
leđa leđ	325/0/0 n2	sladak sladk	454,455/0/438,441
lice lic	325/0/0	slatko2 sladk	0/0/438,441
lišće lišć	325/0/0 n1	ja mn	463/0/0
lozje lozj	325/0/0	ti tb	464/0/0
more mor	325/0/0	mi mn	465/0/0
ognjište ognjišt	325/0/0	vi v	466/0/0
tle tl	326/0/0	on n	467/0/0
vrhnje vrhnj	326/0/0	sebe sb	468/0/0
Zlata Zlat	345/0/0 f1	tko tk	469/0/0
aga ag	345/0/0 m	kojetko kojetk	469/0/0
duka duk	345/0/0 m dužd	moj moj	476/0/441
seka sek	345/0/0	tvoj tvoji	476/0/441
starješina starješin	345/0/0 m	svoj svoj	476/0/441
kolega kolegi	345/0/0 m	naš naš	477/0/441
karioka kariok	345/0/0 mf	vaš vaš	477/0/441

njegov njegov 451,461/0/441  
 jedan jedn 453,452/0/0  
 dva dv 486/0/0  
 obadva obadv 486/0/0  
 oba ob 487/0/0  
 tri tr 488/0/0  
 četiri četir 489/0/0  
 dvoje dvoj 490/0/0  
 oboje oboj 490/0/0  
 troje troj 491/0/0  
 četvero četvr 492/0/0  
 četvoro četvr 492/0/0  
 petero petr 492/0/0  
 petoro petr 492/0/0  
 pričati prič 0/501/0  
 ispričavati ispričav 0/501/0  
 ispričavati\_se ispričav 0/501/0  
 škripati škrip 0/501,540/0  
 škripiti škrip 0/509/0  
 ispričati isprič 0/502/0  
 ispričati\_se isprič 0/502/0  
 krstiti krst 0/503,504/0  
 pokrstiti pokrst 0/505/0  
 gostiti gost 0/506,507/0  
 ugostiti ugost 0/508/0  
 piliti pil 0/509,510/0  
 ispiliti ispil 0/511/0  
 vidjeti vid 0/512,513,514/0  
 zavidjeti zavid 0/512,513/0  
 šumjeti šum 0/512,513/0  
 prešutjeti prešut 0/514/0  
 veljeti vel 0/512,513/0  
 darovati daro 0/515,516/0  
 kazivati kazi 0/515/0  
 bojevati boje 0/515/0  
 kovati ko 0/515/0  
 iskovati isko 0/516/0  
 štovati što 0/515/0  
 ispoštovati ispošto 0/516/0  
 vrištati vrišt 0/517/0  
 zavrištati zavrišt 0/518/0  
 kričati krič 0/519/0  
 zakričati zakrič 0/520/0  
 bojati\_se boj 0/521/0  
 pobojati\_se poboj 0/522/0  
 stajati stoj 0/523/0  
 iskati istk 0/524,526/0  
 zaiskati zaistk 0/525,527/0

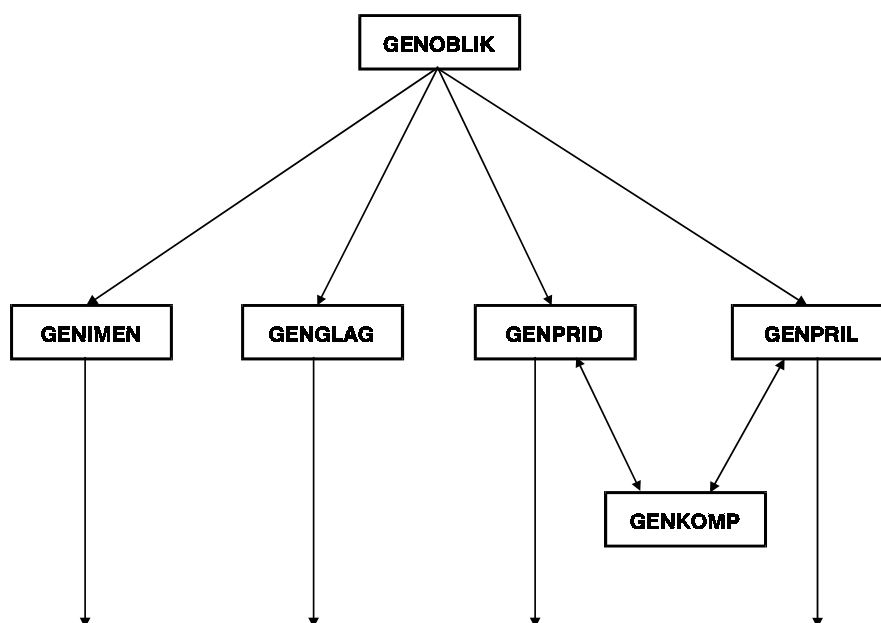
kisnuti kish 0/528/0  
 pokisnuti pokish 0/531,532/0  
 tonuti ton 0/528/0  
 sahnuti sahn 0/528,529/0  
 usahnuti usahn 0/531,532/0  
 raskinuti raskidn 0/530,534/0  
 ogreznuti ogrezn 0/531,532,535/0  
 uvenuti uven 0/531,533/0  
 istrunuti istrun 0/531,533/0  
 nagnuti nagn 0/531,535/0  
 obećavati obećav 0/536/0  
 bljuvati bljuv 0/536/0  
 pobljuvati pobljuv 0/537/0  
 davati dav 0/538/0  
 podavati podav 0/539/0  
 vikati vik 0/540/0  
 povikati povik 0/541/0  
 hripati hrip 0/540/0  
 zahripati zahrip 0/541/0  
 brijati brij 0/542/0  
 izbrijati izbrij 0/543/0  
 sjesti sjedn 0/554/0  
 sresti sretn 0/554/0  
 bosti bod 0/555,557/0  
 zabosti zabod 0/558/0  
 plesti plet 0/559,560/0  
 pući pukn 0/573,574/0  
 leći legn 0/573,575/0  
 nabreći nabrekn 0/573,574/0  
 moći mogn 0/574,577/0  
 pomoći pomogn 0/574/0  
 obući obuk 0/580/0  
 peći pek 0/578,579/0  
 trti tar 0/584,585/0  
 satrti satar 0/586,587/0  
 potrti potar 0/586,587/0  
 ždrijeti ždr 0/589/0  
 mrijeti mr 0/588,589/0  
 umrijeti umr 0/590/0  
 žeti žm 0/597/0  
 prožeti prožm 0/598/0  
 žeti2 žan 0/615,617/0  
 uzeti uzm 0/598/0  
 oteti otm 0/598/0  
 izuti izuj 0/606,608/0  
 obuti obuj 0/606,608/0  
 uspjeti uspěj 0/606/0



## 5.2. Programi

### 5.2.1. Princip rada

Programski se dio sustava sastoji od više međusobno povezanih programa od kojih je jedan — GENOBLIK — zadužen za upravljanje na globalnoj razini. On je zadužen za dohvat osnove i broja (brojeva) uzorka iz leksikona te potom, na osnovi broja uzorka pozivlje odgovarajuće programe — GENIMEN, GENPRID, GENPRIL, GENKOMP, GENGLAG, — koji potom generiraju oblike preobličavanjem osnova i pridruživanjem nastavaka prema uzorku. Međuovisnost programa i smjer obrade podataka prikazana je grafikonom:



Programi su pisani programirnim jezikom *Turbo Pascal 5.5*, a podaci su pohranjeni u podatkovnim datotekama na disku tako da su programi pretežito podatkovno neovisni. Same su matrice tj. popisi nastavaka i transformacija organizirane i ispunjene u programu za obradu datobaza *Access 2.0*, a na kraju su prepisane u obične tekstovne datoteke kojima programi sustava GENOBLIK lako pristupaju.

## 5.2.2. Izvorni kodovi programa

### 5.2.2.1. GENOBLIK

```
{ $M $4000,0,0 }
PROGRAM GenOblik;
{ Program služi kao kontrolni centar za citanje LEXICON.TXT u kojem se
  nalazi popis rijeci s oznakom deklinacijskoga uzorka.
}
USES
  DOS;

CONST
  maxuzoraka = 1000;
  granicnik = ' ';

TYPE
  fonemi      = SET OF CHAR;
  strng       = STRING;
  str5        = STRING[5];
  str4        = STRING[4];
  tipovi_uzorci = ARRAY[1..3,1..4] OF strng;

VAR
  infile, outfile      : TEXT;
  zvucni, bezvucni, samoglasni, suglasni, zvonki      : fonemi;
  tip_uz              : tipovi_uzorci;
  parsed              : ARRAY[1..5] OF strng;
  inime, outime, rijec, osnova, padezi, znacenje, rod, line : strng;
  line5, broj         : str5;
  line3               : str4;
  ch                  : CHAR;
  ln15                : BYTE ABSOLUTE line5;
  ln13                : BYTE ABSOLUTE line3;
  dummy, n, x, y, z   : INTEGER;

PROCEDURE InitVars;
VAR
  i, j : BYTE;
BEGIN { InitVars }
  rijec := '';
  osnova := '';
  znacenje := '';
  rod := '';
  broj := '';
  FOR x := 1 TO 3 DO
    FOR y := 1 TO 4 DO
      tip_uz[x,y] := ' ';
  FOR x := 1 TO 5 DO
    parsed[x] := ' ';
END; { InitVars }

PROCEDURE ParseLine(VAR l:strng; VAR k:BYTE; g:CHAR);
{ funkcija kao rezultat daje broj argumenata u retku l razdvojenih
  granicnikom g. Argumenti su u ARRAY parsed spremijeni po redu }
VAR
  ln1 : BYTE ABSOLUTE l;
  j : BYTE;
BEGIN { ParseLine }
  WHILE l[ln1]=' ' DO { ocisti BLANK iza }
    DEC(ln1);
  WHILE l[1]=' ' DO { ocisti BLANK ispred }
    DELETE(1,1,1);
  k := 1;
  j := POS(g,l);
  WHILE j>0 DO
    BEGIN
      parsed[k] := COPY(1,1,j-1);
      INC(k);
      l := COPY(1,j+1,255);
      WHILE l[1]=' ' DO { ocisti BLANK ispred }
        DELETE(1,1,1);
      j := POS(g,l);
    END;
  parsed[k] := l;
END; { ParseLine }

FUNCTION NadjiRijec(VAR rijec:strng; VAR osnova:strng):BOOLEAN;
{ funkcija otvara rječnik datnicu, traži rijec, uzimlje iz datnice osnovu,
  broj tipova promjene i broj uzoraka u svakom tipu promjene }
VAR
  lex      : TEXT;
  line, t1, t2, t3 : TEXT;
  dummy    : INTEGER;
  nadjeno  : BOOLEAN;
  ln1      : BYTE ABSOLUTE line;
  ln13     : BYTE ABSOLUTE rod;
  i, m     : BYTE;
BEGIN { NadjiRijec }
  nadjeno := FALSE;
  ASSIGN(lex, 'LEXICON.TXT'); RESET(lex);
  REPEAT
    READLN(lex, line);
    i := POS(granicnik, line);
    IF rijec = COPY(line, 1, i-1) THEN
      BEGIN
        nadjeno := TRUE;
        ParseLine(line, i, granicnik);
        osnova := parsed[2];
        line := parsed[3];
        IF i>3 THEN
          rod := parsed[4];
        ELSE
          rod := ' ';
        IF i>4 THEN
          znacenje := parsed[5];
        ELSE
          znacenje := ' ';
        FOR m := 1 TO 5 DO
          parsed[m] := ' ';
        m := 1;
        ParseLine(line, m, '/');
        t1 := parsed[1];
        t2 := parsed[2];
        t3 := parsed[3];
        FOR x := 1 TO 3 DO
          BEGIN
            FOR m := 1 TO 5 DO
              parsed[m] := ' ';
            m := 1;
          END;
        END;
      END;
  UNTIL NOT (i>0);
  IF nadjeno THEN
    rijec := t1;
    osnova := t2;
    znacenje := t3;
    rod := t4;
  END;
```

```

CASE x OF
  1 : ParseLine(t1,m,',');
  2 : ParseLine(t2,m,',');
  3 : ParseLine(t3,m,',');
END;
FOR y := 1 TO 4 DO
  tip_uz[x,y] := parsed[y];
END
END
UNTIL EOF(lex) OR nadjeno;
NadjiriJec := nadjeno;
CLOSE(lex)
END; { NadjiriJec }

BEGIN { GenOblik }
WRITELN(MEMAVAIL);
REPEAT
  InitVars;
  IF PARAMCOUNT<1 THEN
    REPEAT
      WRITE('Unesi rije~ : '); READLN(riJec)
    UNTIL riJec<>' '
  ELSE
    riJec := PARAMSTR(1);
  IF NadjiriJec(riJec,osnova) THEN
    BEGIN
      FOR x := 1 TO 3 DO { *** od 1 do 3 tipa promjene *** }
        FOR y := 1 TO 4 DO { *** od 1 do max 4 uzorka u tipu *** }
          IF tip_uz[x,y]<>' ' THEN
            BEGIN
              line := osnova+' '+tip_uz[x,y];
              VAL(tip_uz[x,y],n,dummy);
              CASE n OF
                1..404 : BEGIN
                  line := line+' '+rod+' '+znacenje;
                  EXEC('c:\jezici\tp\dr\genimen.exe',line);
                END;
                431..442 : IF tip_uz[1,1]='0' THEN
                  BEGIN
                    line := riJec+' '+line;
                    EXEC('c:\jezici\tp\dr\genpril.exe',line)
                  END
                ELSE
                  EXEC('c:\jezici\tp\dr\genkomp.exe',line);
                451..492 : EXEC('c:\jezici\tp\dr\genprid.exe',line);
                500..900 : BEGIN
                  line := riJec+' '+line;
                  EXEC('c:\jezici\tp\dr\genglag.exe',line)
                END
              END
            END
          END
        END
      END
    END
  ELSE
    WRITELN('Ne poznajem tu rije~!',CHR(7))
  UNTIL FALSE OR (PARAMCOUNT>0)
END. { GenOblik }

```

## 5.2.2.2. GENIMEN

```

PROGRAM GenImen;
{ Program generira oblike paradigme imenica. U NASTIMN.TXT nalazi se
  popis svih nastavaka u svim uzorcima. U LEXICON4.TXT nalazi se popis
  rije~i s oznakom deklinacijskoga uzorka. U TRNSIMN.TXT nalazi se popis
  transformacija osnova u svim uzorcima za svaki padez.
}
CONST
  maxuzoraka = 404;
  granicnik = ' ';
TYPE
  fonemi = SET OF CHAR;
  str = STRING;
  str5 = STRING[5];
  str4 = STRING[4];
  arr5 = ARRAY[1..maxuzoraka,1..14] OF str5;
  arr4 = ARRAY[1..maxuzoraka,1..14] OF str4;
  arr5ptr = ^arr5;
  arr4ptr = ^arr4;
  prdgm = ARRAY[1..4,1..14] OF str;
VAR
  infile, outfile : TEXT;
  zvucni, bezvucni, samoglasni, suglasni, zvonki : fonemi;
  un1, un2, un3, un4 : arr5ptr;
  to1, to2, to3, to4 : arr4ptr;
  osobitosti : ARRAY[1..maxuzoraka] OF BYTE;
  paradigme : prdgm;
  inime, outime, riJec, osnova, padezi, znacenje, rod : str;
  line5, broj : str5;
  line3 : str4;
  ch : CHAR;
  lnr : BYTE ABSOLUTE rod;
  ln15 : BYTE ABSOLUTE line5;
  ln13 : BYTE ABSOLUTE line3;
  uzorak, nprdgm, n, x, y, z : INTEGER;

PROCEDURE InitVars;
VAR
  i, j : BYTE;
BEGIN { InitVars }
  zvucni := ['c','b','d','g','z','t','e'];
  bezvucni := ['c','h','k','p','s','t','l','n','m'];
  samoglasni := ['a','e','i','o','u'];
  zvonki := ['f','l','m','n','r','v','w','x'];
  suglasni := zvucni + bezvucni;
  padezi := 'NGDAVLI';
  riJec := '';
  osnova := '';
  znacenje := '';
  uzorak := 0;
  rod := '';
  broj := '';
  FOR i := 1 TO 4 DO
    FOR j := 1 TO 14 DO
      paradigme[i,j] := ''
    END
  END; { InitVars }

PROCEDURE InitUzorke;
{ Procedura otvara datnicu NASTIMN.TXT s nastavcima, učitava ih i smjesta
  u ARRAY un1. Ukoliko postoje alternativni oblici u paradigmi, ukupan se
  broj paradigmi nalazi u osobitosti[x]. Ti se podaci također smjestaju u
  odgovarajuće unX arrays. Datnica nastavci ima ovakvu strk:
  broj uzorka:3, broj starog uzorka:3, n_jd, g_jd, d_jd, a_jd, v_jd, l_jd,
  i_jd, n_mn, g_mn, d_mn, a_mn, v_mn, l_mn, i_mn (svi :5),

```

```

n_jd2, g_jd2 itd., broj_paradigmi:1 }
BEGIN { InitUzorke }

{ *** Rezervirati mjesto za unX arrays *** }
NEW(un1);
NEW(un2);
NEW(un3);
NEW(un4);

{ *** BLANK u sve nastavke *** }
FOR x := 1 TO maxuzoraka DO
  FOR y := 1 TO 14 DO
    BEGIN
      un1^[x,y] := ' ';
      un2^[x,y] := ' ';
      un3^[x,y] := ' ';
      un4^[x,y] := ' ';
    END;

{ *** Ispuni uzorke nastavcima *** }
  ASSIGN(infile,'NASTIMN.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line5);
          WHILE line5[lnl5]=' ' DO
            DEC(line5[0]);
          un1^[x,y] := line5;
        END;
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line5);
          WHILE line5[lnl5]=' ' DO
            DEC(line5[0]);
          un2^[x,y] := line5;
        END;
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line5);
          WHILE line5[lnl5]=' ' DO
            DEC(line5[0]);
          un3^[x,y] := line5;
        END;
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line5);
          WHILE line5[lnl5]=' ' DO
            DEC(line5[0]);
          un4^[x,y] := line5;
        END;
      READ(infile,ch);
      osobitosti[x] := ORD(ch)-48;
      READLN(infile);
    END;
  CLOSE(infile);

{ *** ispis nastavaka u datnicu za probu
  ASSIGN(outfile,'PROBAL.TXT'); REWRITE(outfile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      WRITE(outfile,x:6);
      FOR y := 1 TO 14 DO
        WRITE(outfile,un1^[x,y]:5);
      FOR y := 1 TO 14 DO
        WRITE(outfile,un2^[x,y]:5);
      FOR y := 1 TO 14 DO
        WRITE(outfile,un3^[x,y]:5);
      FOR y := 1 TO 14 DO
        WRITE(outfile,un4^[x,y]:5);
      WRITE(outfile,osobitosti[x]:1);
    END;
  CLOSE(outfile);
}
END; { InitUzorke }

PROCEDURE InitPreoblike;
{ Procedura otvara datnicu TRNSIMN.TXT s propisima o preoblikama osnova,
  učitava ih i smjesta u ARRAY tol. Ukoliko postoje alternativni oblici u
  paradigmi, ukupan se broj paradigmi nalazi u osobitosti[x]. Ti se podaci
  također smjesta u odgovarajuće toX arrays. Datnica TRNSIMN.TXT ima ovakvu
  strk: broj_uzorka:3, broj_starog_uzorka:3, n_jd, g_jd, d_jd, a_jd, v_jd, l_jd,
  i_jd, n_mn, g_mn, d_mn, a_mn, v_mn, l_mn, i_mn (svi :3),
  n_jd2, g_jd2 itd., broj_paradigmi:1 }-
BEGIN { InitPreoblike }

{ *** Rezervirati mjesto za toX arrays *** }
NEW(tol);
NEW(to2);
NEW(to3);
NEW(to4);

{ *** BLANK u sve nastavke *** }
FOR x := 1 TO maxuzoraka DO
  FOR y := 1 TO 14 DO
    BEGIN
      tol^[x,y] := ' ';
      to2^[x,y] := ' ';
      to3^[x,y] := ' ';
      to4^[x,y] := ' ';
    END;

{ *** Ispuni uzorke preoblikama *** }
  ASSIGN(infile,'TRNSIMN.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line3);
          WHILE line3[lnl3]=' ' DO
            DEC(line3[0]);
          tol^[x,y] := line3;
        END;
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line3);
          WHILE line3[lnl3]=' ' DO
            DEC(line3[0]);
          to2^[x,y] := line3;
        END;
      FOR y := 1 TO 14 DO
        BEGIN
          READ(infile,line3);
          WHILE line3[lnl3]=' ' DO
            DEC(line3[0]);

```

```

        to3^[x,y] := line3
    END;
    FOR y := 1 TO 14 DO
        BEGIN
            READ(infile,line3);
            WHILE line3[lnl3]=' ' DO
                DEC(line3[0]);
                to4^[x,y] := line3
            END;
            READLN(infile,ch)
        END;
    CLOSE(infile);
}
{ *** ispis nastavaka u datnicu za probu
  ASSIGN(outfile,'PROBA2.TXT'); REWRITE(outfile);
  FOR x := 1 TO maxuzoraka DO
      BEGIN
          WRITE(outfile,x:6);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to1^[x,y]:5);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to2^[x,y]:5);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to3^[x,y]:5);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to4^[x,y]:5);
          WRITELN(outfile,osobitosti[x]:1)
        END;
      CLOSE(outfile)
    }
  END; { InitPreoblake }

PROCEDURE DzLjNj(VAR s:str; smjer:BOOLEAN);
{ procedura preoblicuje sve d, lj i nj u stringu s u š (#235), æ (#156) i
  þ (#252) kad je smjer FALSE. Uz smjer TRUE i smjer je pretvorbe obrnut }
VAR
  i : BYTE;
BEGIN
  { DzLjNj }
  CASE smjer OF
    FALSE : BEGIN
        REPEAT
            i := POS('d',s);
            IF i>0 THEN
                BEGIN
                    DELETE(s,i,1);
                    s[i] := 'š';
                END
            UNTIL i=0;
        REPEAT
            i := POS('lj',s);
            IF i>0 THEN
                BEGIN
                    DELETE(s,i,1);
                    s[i] := 'æ';
                END
            UNTIL i=0;
        REPEAT
            i := POS('nj',s);
            IF i>0 THEN
                BEGIN
                    DELETE(s,i,1);
                    s[i] := 'þ';
                END
            UNTIL i=0;
        TRUE : BEGIN
            REPEAT
                i := POS('š',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'd';
                        INSERT('.',s,i+1)
                    END
                UNTIL i=0;
            REPEAT
                i := POS('æ',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'lj';
                        INSERT('j',s,i+1)
                    END
                UNTIL i=0;
            REPEAT
                i := POS('þ',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'nj';
                        INSERT('j',s,i+1)
                    END
                UNTIL i=0;
            END
        END
    END
  END; { DzLjNj }

PROCEDURE UmetniA(VAR osn:str);
{ procedura umeće 'nepostojano' a ispred zadnjega pismena }
BEGIN { UmetniA }
  INSERT('a',osn,LENGTH(osn))
END; { UmetniA }

PROCEDURE Sibilizacija(VAR osn:str);
VAR
  i : BYTE ABSOLUTE osn;
BEGIN { Sibilizacija }
  CASE osn[i] OF
    'g' : osn[i] := 'z';
    'n' : osn[i] := 's';
    'k' : osn[i] := 'c';
  END
END; { Sibilizacija }

PROCEDURE Palatalizacija(VAR osn:str);
VAR
  i : BYTE ABSOLUTE osn;
BEGIN { Palatalizacija }
  CASE osn[i] OF
    'c' : osn[i] := 'č';
    'g' : osn[i] := 'đ';
    'h' : osn[i] := 'j';
    'k' : osn[i] := 'ć';
    'z' : osn[i] := 'ž';
  END
END; { Palatalizacija }

PROCEDURE Bipalatalizacija(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
  i : BYTE;
BEGIN { Bipalatalizacija }
  CASE osn[lnosn] OF
    'k' : IF ((osn[lnosn-2]='s') AND (osn[lnosn-1]='t')) OR
          ((osn[lnosn-2]='z') AND (osn[lnosn-1]='d')) THEN
        osn := COPY(osn,1,lnosn-3)+'č';
    'l','n' : CASE osn[lnosn-1] OF

```

```

        's' : osn[lnosn-1] := '{';
        'z' : osn[lnosn-1] := '{';
    END;
    '}', '~' : CASE osn[lnosn-1] OF
        's' : osn[lnosn-1] := '{';
        'z' : osn[lnosn-1] := '{';
        '}' : osn[lnosn-1] := '{';
    END
END
END; { Bipalatalizacija }

PROCEDURE Jotacija(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { Jotacija }
    CASE osn[lnosn] OF
        'd' : osn[lnosn] := 'j';
        't' : osn[lnosn] := 'j';
    END
END; { Jotacija }

PROCEDURE VokaliziranjeL(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { VokaliziranjeL }
    IF osn[lnosn-1]='1' THEN
        osn[lnosn-1]:='o'
    ELSE
        BEGIN
            IF osn[lnosn]='1' THEN
                osn[lnosn]:='o'
            END
        END
    END; { VokaliziranjeL }

PROCEDURE OdbaciKonacniC(VAR osn:str);
BEGIN { OdbaciKonacniC }
    DEC(osn[0])
END; { OdbaciKonacniC }

PROCEDURE Obezvucavanje(VAR osn:str);
VAR
    nadjeno : BOOLEAN;
    i : BYTE;
BEGIN { Obezvucavanje }
    nadjeno := FALSE;
    i := LENGTH(osn);
    REPEAT
        IF (osn[i-1] IN zvucni) AND (osn[i] IN bezvucni) THEN
            nadjeno := TRUE
        ELSE
            DEC(i)
        UNTIL nadjeno OR (i=1);
    IF nadjeno THEN
        CASE osn[i-1] OF
            'a' : osn[i-1] := '{';
            'b' : osn[i-1] := 'p';
            'd' : osn[i-1] := 't';
            'g' : osn[i-1] := 'k';
            'z' : osn[i-1] := 's';
            'l' : osn[i-1] := 'j';
            'e' : osn[i-1] := '~';
        END
    END; { Obezvucavanje }

PROCEDURE IzbaciUnutarnjiC(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { IzbaciUnutarnjiC }
    CASE osn[lnosn] OF
        'c' : IF osn[lnosn-1] IN ['d','t'] THEN
            DELETE(osn,lnosn-1,1);
        'k' : BEGIN
            IF osn[lnosn-1]='t' THEN
                CASE osn[lnosn-2] OF
                    's','{' : DELETE(osn,lnosn-1,1);
                END
            ELSE
                BEGIN
                    IF osn[lnosn-1]='d' THEN
                        CASE osn[lnosn-2] OF
                            'z','{' : BEGIN
                                DELETE(osn,lnosn-1,1);
                                Obezvucavanje(osn)
                            END;
                        END
                    END
                END
            END;
        'l' : IF (osn[lnosn-1]='t') AND (osn[lnosn-2]='s') THEN
            DELETE(osn,lnosn-1,1)
        END
    END; { IzbaciUnutarnjiC }

PROCEDURE ProsiriSJ(VAR osn:str);
BEGIN { ProsiriSJ }
    osn := osn + 'j'
END; { ProsiriSJ }

PROCEDURE KratiJatUI(VAR osn:str);
VAR
    i : BYTE;
BEGIN { KratiJatUI }
    i := POS('ije',osn);
    IF i>0 THEN
        DELETE(osn,i+1,2)
    ELSE
        BEGIN
            i := POS('je',osn);
            IF i>0 THEN
                BEGIN
                    DELETE(osn,i,1);
                    osn[i] := 'i';
                END
            END
        END
    END; { KratiJatUI }

PROCEDURE ProsiriSEr(VAR osn:str);
BEGIN { Prosiri s -er- }
    osn := osn + 'er'
END; { Prosiri s -er- }

PROCEDURE StegniV(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniV }
    IF (osn[lnosn] IN samoglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
        DEC(osn[0])
    ELSE
        BEGIN
            IF (osn[lnosn-1] IN samoglasni) AND (osn[lnosn-2]=osn[lnosn-1])
                THEN DELETE(osn,lnosn-1,1)
        END
    END
END

```

```

END
END; { StegniV }

PROCEDURE StegniC(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniC }
  IF (osn[lnosn] IN suglasi) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN suglasi) AND (osn[lnosn-2]=osn[lnosn-1])
      THEN DELETE(osn,lnosn-1,1)
    END
  END
END; { StegniC }

PROCEDURE ObezvucavanjeBezOkoline(VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i : BYTE;
BEGIN { ObezvucavanjeBezOkoline }
  nadjeno := FALSE;
  i := LENGTH(osn);
  REPEAT
    IF osn[i] IN zvucni THEN
      nadjeno := TRUE
    ELSE
      DEC(i)
    UNTIL nadjeno OR (i=1);
    IF nadjeno THEN
      CASE osn[i] OF
        'b' : osn[i] := 'p';
        'd' : osn[i] := 't';
        'g' : osn[i] := 'k';
        'z' : osn[i] := 's';
        'l' : osn[i] := 'r';
        'e' : osn[i] := '~';
      END
    END
  END; { ObezvucavanjeBezOkoline }

PROCEDURE Umetnia2(VAR osn:str);
{ procedura umece 'nepostojano' a ispred predzadnjega pismena }
BEGIN { Umetnia }
  INSERT('a',osn,LENGTH(osn)-1)
END; { Umetnia }

PROCEDURE SrediJat(VAR osn:str);
VAR
  i : BYTE;
BEGIN { SrediJat }
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      DELETE(osn,i,1);
      IF (osn[i-1]='r') AND (osn[i-2] IN suglasi+zvonki) THEN
        INSERT('e',osn,i)
      ELSE
        INSERT('je',osn,i)
      END
    END
  END; { SrediJat }

FUNCTION PreoblikaOsn(osnova:str; uzorak:INTEGER; paradigm,padez:BYTE):str;
{ funkcija preoblicuje osnovu u skladu sa svakim padezom svake paradigme
svakoga uzorka prema uputama pohranjenim u TRNSIMN.TXT }
VAR
  preoblike : str4;
  lnpreoblike : BYTE ABSOLUTE preoblike;
  i : BYTE;
BEGIN { PreoblikaOsn }
  DzLjNj(osnova,FALSE);
  CASE paradigm OF
    1 : BEGIN
      preoblike := tol^[uzorak,padez];
      FOR x := 1 TO lnpreoblike DO
        CASE preoblike[x] OF
          '1' : Umetnia(osnova);
          '2' : Sibilizacija(osnova);
          '3' : Palatalizacija(osnova);
          '4' : Bipalatalizacija(osnova);
          '5' : Jotacija(osnova);
          '6' : Vokaliziranje(osnova);
          '7' : OdbaciKonacniC(osnova);
          '8' : IzbaciUnutarnjiC(osnova);
          '9' : ProsiriSJ(osnova);
          'a' : Obezvucavanje(osnova);
          'b' : KratiJatUI(osnova);
          'c' : osnova := "-ovjek";
          'd' : ProsiriSEr(osnova);
          'e' : StegniV(osnova);
          'f' : StegniC(osnova);
          'g' : ObezvucavanjeBezOkoline(osnova);
          'h' : Umetnia2(osnova);
          'i' : SrediJat(osnova);
        END
      END;
    2 : BEGIN
      preoblike := to2^[uzorak,padez];
      FOR x := 1 TO lnpreoblike DO
        CASE preoblike[x] OF
          '1' : Umetnia(osnova);
          '2' : Sibilizacija(osnova);
          '3' : Palatalizacija(osnova);
          '4' : Bipalatalizacija(osnova);
          '5' : Jotacija(osnova);
          '6' : Vokaliziranje(osnova);
          '7' : OdbaciKonacniC(osnova);
          '8' : IzbaciUnutarnjiC(osnova);
          '9' : ProsiriSJ(osnova);
          'a' : Obezvucavanje(osnova);
          'b' : KratiJatUI(osnova);
          'c' : ProsiriSEr(osnova);
          'd' : StegniV(osnova);
          'e' : StegniC(osnova);
          'f' : ObezvucavanjeBezOkoline(osnova);
          'g' : Umetnia2(osnova);
          'h' : SrediJat(osnova);
        END
      END;
    3 : BEGIN
      preoblike := to3^[uzorak,padez];
      FOR x := 1 TO lnpreoblike DO
        CASE preoblike[x] OF
          '1' : Umetnia(osnova);
          '2' : Sibilizacija(osnova);
          '3' : Palatalizacija(osnova);
          '4' : Bipalatalizacija(osnova);
          '5' : Jotacija(osnova);
          '6' : Vokaliziranje(osnova);
          '7' : OdbaciKonacniC(osnova);
          '8' : IzbaciUnutarnjiC(osnova);
          '9' : ProsiriSJ(osnova);
          'a' : Obezvucavanje(osnova);
          'b' : KratiJatUI(osnova);
        END
      END;
    END
  END;
END;

```

```

        'e' : StegniV(osnova);
        'f' : StegniC(osnova);
        'g' : ObezvucavanjeBezOkoline(osnova);
        'h' : UmetniA2(osnova);
        'i' : SrediJat(osnova);
    END;
END;
4 : BEGIN
    preoblake := to4^[uzorak,padez];
    FOR x := 1 TO lnpreoblake DO
        CASE preoblake[x] OF
            '1' : UmetniA(osnova);
            '2' : Sibilizacija(osnova);
            '3' : Palatalizacija(osnova);
            '4' : Bipalatalizacija(osnova);
            '5' : Jotacija(osnova);
            '6' : Vokaliziranje(osnova);
            '7' : OdbaciKonacniC(osnova);
            '8' : IzbaciUnutarnjiC(osnova);
            '9' : ProsiriSJ(osnova);
            'a' : Obezvucavanje(osnova);
            'b' : KratiJatUI(osnova);
            'c' : StegniV(osnova);
            'd' : StegniC(osnova);
            'g' : ObezvucavanjeBezOkoline(osnova);
            'h' : UmetniA2(osnova);
            'i' : SrediJat(osnova);
        END;
    END;
END;
DzLjNj(osnova,TRUE);
i := POS('e',osnova);
IF i>0 THEN
    BEGIN
        osnova[i] := 'i';
        INSERT('je',osnova,i+1)
    END;
    PreoblikaOsn := osnova
END; { PreoblikaOsn }

PROCEDURE GenParadigme(osnova:str; uzorak:INTEGER);
VAR
    posnova : str;
    p        : BYTE;
BEGIN { GenParadigme }
    FOR p := 1 TO osobitosti[uzorak] DO
        FOR n := 1 TO 14 DO
            BEGIN
                posnova := PreoblikaOsn(osnova,uzorak,p,n);
                CASE p OF
                    1 : BEGIN
                        rijec := posnova+un1^[uzorak,n];
                        IF un1^[uzorak,n]<>'.' THEN
                            paradigme[p,n] := rijec
                        ELSE
                            paradigme[p,n] := '.'
                        END;
                    2 : BEGIN
                        rijec := posnova+un2^[uzorak,n];
                        IF un2^[uzorak,n]<>'.' THEN
                            paradigme[p,n] := rijec
                        ELSE
                            paradigme[p,n] := ' '
                        END;
                    3 : BEGIN
                        rijec := posnova+un3^[uzorak,n];
                        IF un3^[uzorak,n]<>'.' THEN
                            paradigme[p,n] := rijec
                        ELSE
                            paradigme[p,n] := ' '
                        END;
                    4 : BEGIN
                        rijec := posnova+un4^[uzorak,n];
                        IF un4^[uzorak,n]<>'.' THEN
                            paradigme[p,n] := rijec
                        ELSE
                            paradigme[p,n] := ' '
                        END;
                END; { Case }
                IF (broj='jd') AND (n IN [8..14]) THEN
                    paradigme[p,n] := ' ';
                IF (broj='mn') AND (n IN [1..7]) THEN
                    paradigme[p,n] := ' ';
                END;
            END; { GenParadigme }

PROCEDURE PisiParadigme;
VAR
    p : BYTE;
BEGIN { PisiParadigme }
    WRITELN('Uzorak : ',uzorak,' Rod : ',rod,' Broj : ',broj,' Znacenje : ',znacenje);
    FOR p := 1 TO osobitosti[uzorak] DO
        BEGIN
            FOR n := 1 TO 7 DO
                WRITELN(padezi[n],paradigme[p,n]:20,paradigme[p,n+7]:20);
            WRITELN
        END;
    END; { PisiParadigme }

BEGIN { GenImen }
    WRITELN(MEMAVAIL);
    InitUzorke;
    InitPreoblake;
    WRITELN(MEMAVAIL);
    InitVars;
    osnova := PARAMSTR(1);
    rijec := PARAMSTR(2);
    rod := PARAMSTR(3);
    znacenje := PARAMSTR(4);
    VAL(rijec,uzorak,z);
    { *** sredi rod *** }
    IF rod=' ' THEN
        CASE uzorak OF
            1..318 : rod := 'm';
            319..342 : rod := 'n';
            343..374 : rod := 'p';
            375 : rod := 'm';
            376..397 : rod := 'f';
            398..401 : rod := 'm';
            402 : rod := 'f';
            403..404 : rod := 'n'
        END;
    ELSE
        BEGIN
            IF lnz>1 THEN
                CASE rod[2] OF
                    '1' : BEGIN
                        broj := 'jd';
                        rod := rod[1]
                    END;
                    '2' : BEGIN
                        broj := 'mn';
                        rod := rod[1]
                    END;
                END;
            END;
        END;
    END;

```



```

        END
      ELSE
        broj := 'jd+mn'
      END;
    { *** generiraj paradigme *** }
    GenParadigme (osnova, uzorak);
    PisiParadigme
  END. { GenImen }

```

### 5.2.2.3. GENPRID

```

PROGRAM GenPrId;
{ Program generira oblike paradigme pridjeva. U NASTPRID.TXT nalazi se
  popis svih nastavaka u svim uzorcima. U TRNSPRID.TXT nalazi se popis
  transformacija osnova u svim uzorcima za svaki padez.
}
CONST
  maxuzoraka = 42;
  maxprdgms = 5;
  granicnik = ' ';
TYPE
  fonemi = SET OF CHAR;
  str = STRING;
  str3 = STRING[3];
  str4 = STRING[4];
  arr3 = ARRAY[1..maxuzoraka, 1..3, 1..14] OF str3;
  arr3ptr = ^arr3;
  prdgm = ARRAY[1..maxprdgms, 1..3, 1..14] OF str;
VAR
  infile, outfile : TEXT;
  zvucni, bezzvucni, samoglasni, suglasni, zvonki : fonemi;
  un1, un2, un3, un4, un5, to1, to2, to3, to4, to5 : arr3ptr;
  o : ARRAY[1..maxuzoraka] OF BYTE;
  paradigme : prdgm;
  inime, outime, rijec, osnova, osn, padezi, znacenje, rod : str;
  line3 : str3;
  line4, broj : str4;
  ch : CHAR;
  ln13 : BYTE ABSOLUTE line3;
  ln14 : BYTE ABSOLUTE line4;
  uzorak, nprdgms, n, x, y, z : INTEGER;

PROCEDURE InitVars;
VAR
  i, j, k : BYTE;
BEGIN { InitVars }
  zvucni := ['b', 'd', 'g', 'z', 't', 'e'];
  bezzvucni := ['c', 'h', 'k', 'p', 's', 't', 'f', 'l', 'm', 'n', 'r', 'v', 'œ', 'a'];
  samoglasni := ['a', 'e', 'i', 'o', 'u'];
  zvonki := ['f', 'g', 'l', 'm', 'n', 'r', 'v', 'œ', 'a'];
  suglasni := zvucni + bezzvucni;
  padezi := 'NGDAVLINGDAVLI';
  rijec := ' ';
  osnova := ' ';
  osn := ' ';
  znacenje := ' ';
  uzorak := 0;
  rod := ' ';
  broj := ' ';
END; { InitVars }

PROCEDURE CistiPrdgms;
BEGIN { CistiPrdgms }
  FOR x := 1 TO maxprdgms DO
    FOR y := 1 TO 3 DO
      FOR z := 1 TO 14 DO
        paradigme[x,y,z] := ' '
      END;
    END;
  END; { CistiPrdgms }

PROCEDURE InitUzorke;
{ Procedura otvara datnicu NASTPRID.TXT s nastavcima, učitava ih i smješta
  u ARRAY un1. Ukoliko postoje alternativni oblici u paradigmi, ukupan se
  broj paradigmi nalazi u o[x]. Ti se podaci također smještaju u
  odgovarajuće unX arrays. Datnica nastavci ima ovakvu strukturu:
  broj_uzorka:3, broj_starog_uzorka:3, n_jd, g_jd, d_jd, a_jd, v_jd, l_jd,
  i_jd, n_mn, g_mn, d_mn, a_mn, v_mn, l_mn, i_mn (svi :5),
  n_jd2, g_jd2 itd., Broj_paradigmi:1 }
BEGIN { InitUzorke }
  { *** Rezervirati mjesto za unX arrays *** }
  NEW(un1);
  NEW(un2);
  NEW(un3);
  NEW(un4);
  NEW(un5);

  { *** BLANK u sve nastavke *** }
  FOR x := 1 TO maxuzoraka DO
    FOR y := 1 TO 3 DO
      FOR z := 1 TO 14 DO
        BEGIN
          un1^[x,y,z] := ' ';
          un2^[x,y,z] := ' ';
          un3^[x,y,z] := ' ';
          un4^[x,y,z] := ' ';
          un5^[x,y,z] := ' ';
        END;
      END;
    END;

  { *** Ispuni uzorke nastavcima *** }
  ASSIGN(infile, 'NASTPRID.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, ch);
      FOR y := 1 TO 3 DO
        FOR z := 1 TO 14 DO
          BEGIN
            READ(infile, line3);
            WHILE line3[ln13] = ' ' DO
              DEC(line3[0]);
            un1^[x,y,z] := line3;
          END;
        END;
      END;
      FOR y := 1 TO 3 DO
        FOR z := 1 TO 14 DO
          BEGIN
            READ(infile, line3);
            WHILE line3[ln13] = ' ' DO
              DEC(line3[0]);
            un2^[x,y,z] := line3;
          END;
        END;
      END;
      FOR y := 1 TO 3 DO
        FOR z := 1 TO 14 DO
          BEGIN

```

```

                                READ(infile,line3);
                                WHILE line3[lnl3]=' ' DO
                                    DEC(line3[0]);
                                    un3^[x,y,z] := line3
                                END;
FOR y := 1 TO 3 DO
FOR z := 1 TO 14 DO
BEGIN
    READ(infile,line3);
    WHILE line3[lnl3]=' ' DO
        DEC(line3[0]);
        un4^[x,y,z] := line3
    END;
FOR y := 1 TO 3 DO
FOR z := 1 TO 14 DO
BEGIN
    READ(infile,line3);
    WHILE line3[lnl3]=' ' DO
        DEC(line3[0]);
        un5^[x,y,z] := line3
    END;
    READ(infile,ch);
    o[x] := ORD(ch)-48;
    READLN(infile);
END;
CLOSE(infile);

{ *** ispis nastavaka u datnicu za probu
  ASSIGN(outfile,'PROBA1.TXT'); REWRITE(outfile);
  FOR x := 1 TO maxuzoraka DO
  BEGIN
      WRITE(outfile,x:6);
      FOR y := 1 TO 14 DO
          WRITE(outfile,un1^[x,y]:5);
      FOR y := 1 TO 14 DO
          WRITE(outfile,un2^[x,y]:5);
      FOR y := 1 TO 14 DO
          WRITE(outfile,un3^[x,y]:5);
      FOR y := 1 TO 14 DO
          WRITE(outfile,un4^[x,y]:5);
      WRITE(outfile,un5^[x,y]:5);
      WRITELN(outfile,o[x]:1)
  END;
  CLOSE(outfile)
}
END; { InitUzorke }

PROCEDURE InitPreoblike;
{ Procedura otvara datnicu TRNSPRID.TXT s propisima o preoblikama osnovna,
  učitava ih i smješta u ARRAY tol. Ukoliko postoje alternativni oblici u
  paradigmi, ukupan se broj paradigmi nalazi u o[x]. Ti se podaci
  također smještaju u odgovarajuće toX arrays. Datnica TRNSPRID.TXT ima ovakvu
  strk: broj uzorka:3, Broj starog uzorka:3, n_jd, g_jd, d_jd, a_jd, v_jd, l_jd,
  i_jd, n_mn, g_mn, d_mn, a_mn, v_mn, l_mn, i_mn (svi :3),
  n_jd2, g_jd2 itd., Broj paradigmi:1 }
BEGIN { InitPreoblike }

    { *** Rezervirati mjesto za toX arrays *** }
    NEW(tol);
    NEW(to2);
    NEW(to3);
    NEW(to4);
    NEW(to5);

    { *** BLANK u sve nastavke *** }
    FOR x := 1 TO maxuzoraka DO
        FOR y := 1 TO 3 DO
            FOR z := 1 TO 14 DO
                BEGIN
                    tol^[x,y,z] := ' ';
                    to2^[x,y,z] := ' ';
                    to3^[x,y,z] := ' ';
                    to4^[x,y,z] := ' ';
                    to5^[x,y,z] := ' ';
                END;
            END;
        END;

    { *** Ispuni uzorke preoblikama *** }
    ASSIGN(infile,'TRNSPRID.TXT'); RESET(infile);
    FOR x := 1 TO maxuzoraka DO
        BEGIN
            READ(infile,ch);
            READ(infile,ch);
            READ(infile,ch);
            FOR y := 1 TO 3 DO
                FOR z := 1 TO 14 DO
                    BEGIN
                        READ(infile,line3);
                        WHILE line3[lnl3]=' ' DO
                            DEC(line3[0]);
                            tol^[x,y,z] := line3
                        END;
                    END;
                FOR y := 1 TO 3 DO
                    FOR z := 1 TO 14 DO
                        BEGIN
                            READ(infile,line3);
                            WHILE line3[lnl3]=' ' DO
                                DEC(line3[0]);
                                to2^[x,y,z] := line3
                            END;
                        END;
                    FOR y := 1 TO 3 DO
                        FOR z := 1 TO 14 DO
                            BEGIN
                                READ(infile,line3);
                                WHILE line3[lnl3]=' ' DO
                                    DEC(line3[0]);
                                    to3^[x,y,z] := line3
                                END;
                            END;
                        FOR y := 1 TO 3 DO
                            FOR z := 1 TO 14 DO
                                BEGIN
                                    READ(infile,line3);
                                    WHILE line3[lnl3]=' ' DO
                                        DEC(line3[0]);
                                        to4^[x,y,z] := line3
                                    END;
                                END;
                            FOR y := 1 TO 3 DO
                                FOR z := 1 TO 14 DO
                                    BEGIN
                                        READ(infile,line3);
                                        WHILE line3[lnl3]=' ' DO
                                            DEC(line3[0]);
                                            to5^[x,y,z] := line3
                                        END;
                                    END;
                                READLN(infile,ch)
                            END;
                        END;
                    END;
                END;
            END;
        END;
        CLOSE(infile);

    { *** ispis nastavaka u datnicu za probu
      ASSIGN(outfile,'PROBA2.TXT'); REWRITE(outfile);
      FOR x := 1 TO maxuzoraka DO
      BEGIN
          WRITE(outfile,x:6);
          FOR y := 1 TO 14 DO
              WRITE(outfile,tol^[x,y]:5);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to2^[x,y]:5);
          FOR y := 1 TO 14 DO
              WRITE(outfile,to3^[x,y]:5);
          FOR y := 1 TO 14 DO

```

```

        WRITE(outfile,to4^[x,y]:5);
        WRITELN(outfile,o[x]:1)
    END;
    CLOSE(outfile)
}
END; { InitPreoblike }

PROCEDURE DzLjNj(VAR s:str; smjer:BOOLEAN);
{ procedura preoblicuje sve d, l, j i nj u stringu s u ě (#235), æ (#156) i
n (#252) kad je smjer FALSE. Uz smjer TRUE i smjer je pretvorbe obrnut }
VAR
    i : BYTE;
BEGIN { DzLjNj }
    CASE smjer OF
        FALSE : BEGIN
            REPEAT
                i := POS('d',s);
                IF i>0 THEN
                    BEGIN
                        DELETE(s,i,1);
                        s[i] := 'ě';
                    END
                UNTIL i=0;
            REPEAT
                i := POS('lj',s);
                IF i>0 THEN
                    BEGIN
                        DELETE(s,i,1);
                        s[i] := 'æ';
                    END
                UNTIL i=0;
            REPEAT
                i := POS('nj',s);
                IF i>0 THEN
                    BEGIN
                        DELETE(s,i,1);
                        s[i] := 'n';
                    END
                UNTIL i=0;
            END;
        TRUE : BEGIN
            REPEAT
                i := POS('ě',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'd';
                        INSERT('.',s,i+1)
                    END
                UNTIL i=0;
            REPEAT
                i := POS('æ',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'l';
                        INSERT('j',s,i+1)
                    END
                UNTIL i=0;
            REPEAT
                i := POS('n',s);
                IF i>0 THEN
                    BEGIN
                        s[i] := 'nj';
                        INSERT('j',s,i+1)
                    END
                UNTIL i=0;
            END
        END
    END
END; { DzLjNj }

PROCEDURE UmetniA(VAR osn:str);
{ procedura umeće 'nepostojano' a ispred zadnjega pismena }
BEGIN { UmetniA }
    INSERT('a',osn,LENGTH(osn))
END; { UmetniA }

PROCEDURE Sibilizacija(VAR osn:str);
VAR
    i : BYTE;
BEGIN { Sibilizacija }
    CASE osn[i] OF
        'g' : osn[i] := 'z';
        'h' : osn[i] := 's';
        'k' : osn[i] := 'c';
    END
END; { Sibilizacija }

PROCEDURE Palatalizacija(VAR osn:str);
VAR
    i : BYTE;
BEGIN { Palatalizacija }
    CASE osn[i] OF
        'c' : osn[i] := 'č';
        'g' : osn[i] := 'j';
        'n' : osn[i] := 'nj';
        'k' : osn[i] := 'č';
        'z' : osn[i] := 'ž';
    END
END; { Palatalizacija }

PROCEDURE Bipalatalizacija(VAR osn:str);
VAR
    lnosn : BYTE;
    i : BYTE;
BEGIN { Bipalatalizacija }
    CASE osn[lnosn] OF
        'k' : IF ((osn[lnosn-2]='s') AND (osn[lnosn-1]='t')) OR
                ((osn[lnosn-2]='z') AND (osn[lnosn-1]='d')) THEN
            osn := COPY(osn,1,lnosn-3)+'č';
        'l','n' : CASE osn[lnosn-1] OF
            's' : osn[lnosn-1] := 'š';
            'z' : osn[lnosn-1] := 'ž';
        END;
        't' : CASE osn[lnosn-1] OF
            '~' : osn[lnosn-1] := 'č';
        END;
        'j','~' : CASE osn[lnosn-1] OF
            's' : osn[lnosn-1] := 'š';
            'z' : osn[lnosn-1] := 'ž';
            '~' : osn[lnosn-1] := 'č';
        END
    END
END; { Bipalatalizacija }

PROCEDURE Jotacija(VAR osn:str);
VAR
    lnosn : BYTE;
BEGIN { Jotacija }
    CASE osn[lnosn] OF
        'd' : osn[lnosn] := 'đ';
        't' : osn[lnosn] := 't';
    END
END; { Jotacija }

```

```

PROCEDURE VokaliziranjeL (VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { VokaliziranjeL }
  IF osn[lnosn-1]='i' THEN
    osn[lnosn-1]:='o'
  ELSE
    BEGIN
      IF osn[lnosn]='i' THEN
        osn[lnosn]:='o'
      END
    END
  END; { VokaliziranjeL }

PROCEDURE OdbaciKonacniC (VAR osn:str);
BEGIN { OdbaciKonacniC }
  DEC(osn[0])
END; { OdbaciKonacniC }

PROCEDURE Obezvucavanje (VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i : BYTE;
BEGIN { Obezvucavanje }
  nadjeno := FALSE;
  i := LENGTH(osn);
  REPEAT
    IF (osn[i-1] IN zvucni) AND (osn[i] IN bezvucni) THEN
      nadjeno := TRUE
    ELSE
      DEC(i)
  UNTIL nadjeno OR (i=1);
  IF nadjeno THEN
    CASE osn[i-1] OF
      'i' : osn[i-1] := 'o';
      'b' : osn[i-1] := 'p';
      'd' : osn[i-1] := 't';
      'g' : osn[i-1] := 'k';
      'z' : osn[i-1] := 's';
      'l' : osn[i-1] := 'j';
      'e' : osn[i-1] := 'o';
    END
  END; { Obezvucavanje }

PROCEDURE IzbaciUnutarnjiC (VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { IzbaciUnutarnjiC }
  CASE osn[lnosn] OF
    'c' : IF osn[lnosn-1] IN ['d','t'] THEN
      DELETE(osn,lnosn-1,1);
    'k' : BEGIN
      IF osn[lnosn-1]='t' THEN
        CASE osn[lnosn-2] OF
          's','l' : DELETE(osn,lnosn-1,1);
        END
      ELSE
        BEGIN
          IF osn[lnosn-1]='d' THEN
            CASE osn[lnosn-2] OF
              'z','l' : BEGIN
                DELETE(osn,lnosn-1,1);
                Obezvucavanje(osn)
              END;
            END
          END;
        END
      END;
    'l' : IF (osn[lnosn-1]='t') AND (osn[lnosn-2]='s') THEN
      DELETE(osn,lnosn-1,1);
    'n' : CASE osn[lnosn-1] OF
      'd' : IF osn[lnosn-2]='j' THEN
        DELETE(osn,lnosn-1,1);
      't' : IF osn[lnosn-2]='s' THEN
        DELETE(osn,lnosn-1,1)
      END
    END
  END
END; { IzbaciUnutarnjiC }

PROCEDURE ProsiriSJ (VAR osn:str);
BEGIN { ProsiriSJ }
  osn := osn + 'j'
END; { ProsiriSJ }

PROCEDURE KratiJatUI (VAR osn:str);
VAR
  i : BYTE;
BEGIN { KratiJatUI }
  i := POS('ije',osn);
  IF i>0 THEN
    DELETE(osn,i+1,2)
  ELSE
    BEGIN
      i := POS('je',osn);
      IF i>0 THEN
        BEGIN
          DELETE(osn,i,1);
          osn[i] := 'i'
        END
      END
    END
  END; { KratiJatUI }

PROCEDURE ProsiriSEr (VAR osn:str);
BEGIN { ProsiriSEr }
  osn := osn + 'er'
END; { ProsiriSEr }

PROCEDURE StegniV (VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniV }
  IF (osn[lnosn] IN samoglasni) AND (osn[lnosn]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN samoglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
  END; { StegniV }

PROCEDURE StegniC (VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniC }
  IF (osn[lnosn] IN suglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN suglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
  END; { StegniC }

```

```

PROCEDURE ObezvucavanjeBezOkoline(VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i : BYTE;
BEGIN { ObezvucavanjeBezOkoline }
  nadjeno := FALSE;
  i := LENGTH(osn);
  REPEAT
    IF osn[i] IN zvucni THEN
      nadjeno := TRUE
    ELSE
      DEC(i)
  UNTIL nadjeno OR (i=1);
  IF nadjeno THEN
    CASE osn[i] OF
      'a' : osn[i] := 't';
      'b' : osn[i] := 'p';
      'd' : osn[i] := 't';
      'g' : osn[i] := 'k';
      'z' : osn[i] := 's';
      'l' : osn[i] := 'l';
      'e' : osn[i] := 'u';
    END
  END; { ObezvucavanjeBezOkoline }

PROCEDURE UmetniA2(VAR osn:str);
{ procedura umece prijevoini e ispred predzadnjega pismena }
BEGIN { UmetniA }
  INSERT('a',osn,LENGTH(osn)-1)
END; { UmetniA }

PROCEDURE SrediJat(VAR osn:str);
VAR
  i : BYTE;
BEGIN { SrediJat }
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      DELETE(osn,i,1);
      IF (osn[i-1]='r') AND (osn[i-2] IN suglasi+zvonki) THEN
        INSERT('e',osn,i)
      ELSE
        INSERT('je',osn,i)
    END
  END; { SrediJat }

PROCEDURE PrijevojPuninaE(VAR osn:str);
{ procedura umece prijevoini e ispred zadnjega pismena }
BEGIN { PrijevojPuninaE }
  INSERT('e',osn,LENGTH(osn))
END; { PrijevojPuninaE }

PROCEDURE PrijevojPuninaO(VAR osn:str);
{ procedura umece prijevoini o ispred zadnjega pismena }
BEGIN { PrijevojPuninaO }
  INSERT('o',osn,LENGTH(osn))
END; { PrijevojPuninaO }

PROCEDURE OdbaciPretkonacniC(VAR osn:str);
{ procedura izbacuje pretkonacni C }
BEGIN { OdbaciPretkonacniC }
  DELETE(osn,LENGTH(osn)-1,1)
END; { OdbaciPretkonacniC }

FUNCTION PreoblikaOsn(osnova:str; uzorak:INTEGER; paradigm,rod,padez:BYTE):str;
{ funkcija preoblicuje osnovu u skladu sa svakim padezom svake paradigme
  svakoga uzorka prema uputama pohranjenim u TRNSPRID.TXT }
VAR
  preoblike : str3;
  lnpreoblike : BYTE ABSOLUTE preoblike;
  i : BYTE;
BEGIN { PreoblikaOsn }
  DZLjNj(osnova,FALSE);
  CASE paradigm OF
    1 : BEGIN
      preoblike := to1^[uzorak,rod,padez];
      FOR x := 1 TO lnpreoblike DO
        CASE preoblike[x] OF
          '1' : UmetniA(osnova);
          '2' : Sibilizacija(osnova);
          '3' : Palatalizacija(osnova);
          '4' : Bipalatalizacija(osnova);
          '5' : Jotacija(osnova);
          '6' : Vokaliziranje(osnova);
          '7' : OdbaciKonacniC(osnova);
          '8' : IzbaciUnutarnjiC(osnova);
          '9' : ProsiriSJ(osnova);
          'a' : Obezvucavanje(osnova);
          'b' : KratiJatUI(osnova);
          'c' : osnova := "-ovjek";
          'd' : ProsiriSEr(osnova);
          'e' : StegniV(osnova);
          'f' : StegniC(osnova);
          'g' : ObezvucavanjeBezOkoline(osnova);
          'h' : UmetniA2(osnova);
          'i' : SrediJat(osnova);
          'j' : ;
          'k' : PrijevojPuninaE(osnova);
          'l' : PrijevojPuninaO(osnova);
          'm' : OdbaciPretkonacniC(osnova);
        END
      END;
    2 : BEGIN
      preoblike := to2^[uzorak,rod,padez];
      FOR x := 1 TO lnpreoblike DO
        CASE preoblike[x] OF
          '1' : UmetniA(osnova);
          '2' : Sibilizacija(osnova);
          '3' : Palatalizacija(osnova);
          '4' : Bipalatalizacija(osnova);
          '5' : Jotacija(osnova);
          '6' : Vokaliziranje(osnova);
          '7' : OdbaciKonacniC(osnova);
          '8' : IzbaciUnutarnjiC(osnova);
          '9' : ProsiriSJ(osnova);
          'a' : Obezvucavanje(osnova);
          'b' : KratiJatUI(osnova);
          'c' : ProsiriSEr(osnova);
          'd' : StegniV(osnova);
          'e' : StegniC(osnova);
          'f' : ObezvucavanjeBezOkoline(osnova);
          'g' : UmetniA2(osnova);
          'h' : SrediJat(osnova);
          'i' : ;
          'j' : ;
          'k' : PrijevojPuninaE(osnova);
          'l' : PrijevojPuninaO(osnova);
          'm' : OdbaciPretkonacniC(osnova);
        END
      END;
    3 : BEGIN
      preoblike := to3^[uzorak,rod,padez];

```

```

FOR x := 1 TO lnpreoblke DO
CASE preoblke[x] OF
'1' : UmetniA(osnova);
'2' : Sibilizacija(osnova);
'3' : Palatalizacija(osnova);
'4' : Bipalatalizacija(osnova);
'5' : Jotacija(osnova);
'6' : Vokaliziranje(osnova);
'7' : OdbaciKonacniC(osnova);
'8' : IzbaciUnutarnjiC(osnova);
'9' : ProsiriSJ(osnova);
'a' : Obezvucavanje(osnova);
'b' : KratiJatUI(osnova);
'e' : StegniV(osnova);
'f' : StegniC(osnova);
'g' : ObezvucavanjeBezOkoline(osnova);
'h' : UmetniA2(osnova);
'i' : SrediJat(osnova);
'j' : ;
'k' : PrijevojPuninaE(osnova);
'l' : PrijevojPuninaO(osnova);
'm' : OdbaciPretkonacniC(osnova)
END;
END;
4 : BEGIN
preoblke := to4^[uzorak,rod,padez];
FOR x := 1 TO lnpreoblke DO
CASE preoblke[x] OF
'1' : UmetniA(osnova);
'2' : Sibilizacija(osnova);
'3' : Palatalizacija(osnova);
'4' : Bipalatalizacija(osnova);
'5' : Jotacija(osnova);
'6' : Vokaliziranje(osnova);
'7' : OdbaciKonacniC(osnova);
'8' : IzbaciUnutarnjiC(osnova);
'9' : ProsiriSJ(osnova);
'a' : Obezvucavanje(osnova);
'b' : KratiJatUI(osnova);
'e' : StegniV(osnova);
'f' : StegniC(osnova);
'g' : ObezvucavanjeBezOkoline(osnova);
'h' : UmetniA2(osnova);
'i' : SrediJat(osnova);
'j' : ;
'k' : PrijevojPuninaE(osnova);
'l' : PrijevojPuninaO(osnova);
'm' : OdbaciPretkonacniC(osnova)
END;
END;
5 : BEGIN
preoblke := to5^[uzorak,rod,padez];
FOR x := 1 TO lnpreoblke DO
CASE preoblke[x] OF
'1' : UmetniA(osnova);
'2' : Sibilizacija(osnova);
'3' : Palatalizacija(osnova);
'4' : Bipalatalizacija(osnova);
'5' : Jotacija(osnova);
'6' : Vokaliziranje(osnova);
'7' : OdbaciKonacniC(osnova);
'8' : IzbaciUnutarnjiC(osnova);
'9' : ProsiriSJ(osnova);
'a' : Obezvucavanje(osnova);
'b' : KratiJatUI(osnova);
'e' : StegniV(osnova);
'f' : StegniC(osnova);
'g' : ObezvucavanjeBezOkoline(osnova);
'h' : UmetniA2(osnova);
'i' : SrediJat(osnova);
'j' : ;
'k' : PrijevojPuninaE(osnova);
'l' : PrijevojPuninaO(osnova);
'm' : OdbaciPretkonacniC(osnova)
END;
END;
END;
DzLjNj(osnova,TRUE);
i := POS('ë',osnova);
IF i>0 THEN
BEGIN
osnova[i] := 'i';
INSERT('je',osnova,i+1)
END;
PreoblikaOsn := osnova
END; { PreoblikaOsn }

PROCEDURE GenParadigme(VAR osnova:str; uzorak:INTEGER);
VAR
posnova : str;
p,r : BYTE;
BEGIN { GenParadigme }
FOR p := 1 TO o[uzorak] DO
FOR r := 1 TO 3 DO
FOR n := 1 TO 14 DO
BEGIN
posnova := PreoblikaOsn(osnova,uzorak,p,r,n);
CASE p OF
1 : BEGIN
rijec := posnova+un1^[uzorak,r,n];
IF un1^[uzorak,r,n]<>'.' THEN
BEGIN
CASE uzorak OF
24,25,33,34 : rijec := rijec + 'god'
END;
paradigme[p,r,n] := rijec
END
ELSE
paradigme[p,r,n] := '.'
END;
2 : BEGIN
rijec := posnova+un2^[uzorak,r,n];
IF un2^[uzorak,r,n]<>'.' THEN
BEGIN
CASE uzorak OF
24,25,33,34 : rijec := rijec + 'god'
END;
paradigme[p,r,n] := rijec
END
ELSE
paradigme[p,r,n] := '.'
END;
3 : BEGIN
rijec := posnova+un3^[uzorak,r,n];
IF un3^[uzorak,r,n]<>'.' THEN
BEGIN
CASE uzorak OF
24,25,33,34 : rijec := rijec + 'god'
END;
paradigme[p,r,n] := rijec
END
ELSE
paradigme[p,r,n] := '.'
END;
4 : BEGIN
rijec := posnova+un4^[uzorak,r,n];
IF un4^[uzorak,r,n]<>'.' THEN
BEGIN

```

```

CASE uzorak OF
  24,25,33,34 : rijec := rijec + 'god'
END;
paradigme[p,r,n] := rijec
END
ELSE
  paradigme[p,r,n] := '.'
END;
5 : BEGIN
  rijec := posnova+un5^[uzorak,r,n];
  IF un5^[uzorak,r,n]<>'.' THEN
    BEGIN
      CASE uzorak OF
        24,25,33,34 : rijec := rijec + 'god'
      END;
      paradigme[p,r,n] := rijec
    END
  ELSE
    paradigme[p,r,n] := '.'
  END
END; { Case }
END
END; { GenParadigme }

PROCEDURE PisiParadigme(u:BYTE; ispis:str);
VAR
  p,r : BYTE;
BEGIN { PisiParadigme }
  WRITELN('Uzorak : ',uzorak,' ',ispis);
  WRITELN;
  FOR p := 1 TO u DO
    BEGIN
      FOR n := 1 TO 14 DO
        WRITELN(padezi[n],paradigme[p,1,n]:20,paradigme[p,2,n]:20,paradigme[p,3,n]:20);
      WRITELN
    END
  END; { PisiParadigme }

BEGIN { GenPrid }
  WRITELN(MEMAVAIL);
  InitUzorke;
  InitPreoblike;
  WRITELN(MEMAVAIL);
  InitVars;
  CistiPrdgm;
  osn := PARAMSTR(1);
  rijec := PARAMSTR(2);
  VAL(rijec,uzorak,z);
  GenParadigme(osn,uzorak-450);
  PisiParadigme(c[uzorak-450], ' ');
  WRITE('>'); READLN
END. { GenPrid }

```

## 5.2.2.4. GENKOMP

```

{$M $4000,0,0}
PROGRAM GenKomp;
{ Program generira preobliku osnovne pridjeva tvoreci komparativnu osnovu.
  U NASTKOMP.TXT nalazi se popis svih nastavaka u svim uzorcima.
  U TRNSKOMP.TXT nalazi se popis transformacija osnova u svim uzorcima
}
USES
  DOS;

CONST
  maxuzoraka = 12;

TYPE
  fonemi = SET OF CHAR;
  str     = STRING[3];
  str3    = STRING[3];
  str4    = STRING[4];
  arr4    = ARRAY[1..maxuzoraka] OF str4;

VAR
  infile, outfile      : TEXT;
  zvucni, bezvucni, samoglasni, suglasni, zvonki : fonemi;
  u, t                 : arr4;
  inime, outime, line, line2, osnova, posnova   : str;
  line3                                           : str3;
  line4, broj                                     : str4;
  ch                                              : CHAR;
  ln13                                           : BYTE ABSOLUTE line3;
  ln14                                           : BYTE ABSOLUTE line4;
  uzorak, n, x, y, z                             : INTEGER;

PROCEDURE InitVars;
VAR
  i, j, k : BYTE;
BEGIN { InitVars }
  zvucni := ['b','d','g','z','t','s','l','m','n','r','v','w','m'];
  bezvucni := ['c','h','k','p','t','s','l','m','n'];
  samoglasni := ['a','e','i','o','u'];
  zvonki := ['f','j','l','m','n','r','v','w','m'];
  suglasni := zvucni + bezvucni;
  line := '';
  osnova := '';
  posnova := '';
  uzorak := 0;
END; { InitVars }

PROCEDURE InitUzorke;
{ Procedura otvara datnicu NASTKOMP.TXT s nastavcima za sve uzorke, učitava
  ih i smješta u ARRAY u. Datnica NASTKOMP.TXT ima ovakvu strk:
  broj_uzorka:3, nastavak:3 }
BEGIN { InitUzorke }
  { *** Isprazni uzorke *** }
  FOR x := 1 TO maxuzoraka DO
    u[x] := ' ';
  { *** Ispuni uzorke nastavcima *** }
  ASSIGN(infile, 'NASTKOMP.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, line3);
      WHILE line3[ln13] = ' ' DO
        DEC(line3[0]);
      u[x] := line3;
    END
  END

```

```

        READLN(infile)
    END;
    CLOSE(infile)
END; { InitUzorke }

PROCEDURE InitPreobliske;
{ Procedura otvara datnicu TRNSKOMP.TXT s preoblikama za sve uzorke učitava
ih i smjesta u ARRAY t. Datnica TRNSKOMP.TXT ima ovakvu strk:
broj_uzorka:3, preobliske:4 }
BEGIN { InitPreobliske }
    { *** Isprazni preobliske *** }
    FOR x := 1 TO maxuzoraka DO
        t[x] := ' ';

    { *** Ispuni preobliske *** }
    ASSIGN(infile, 'TRNSKOMP.TXT'); RESET(infile);
    FOR x := 1 TO maxuzoraka DO
        BEGIN
            READ(infile, ch);
            READ(infile, ch);
            READ(infile, ch);
            READ(infile, line4);
            WHILE line4[lnl4] = ' ' DO
                DEC(line4[0]);
            t[x] := line4;
            READLN(infile)
        END;
    CLOSE(infile)
END; { InitPreobliske }

PROCEDURE DzLjNj(VAR s:str; smjer:BOOLEAN);
{ procedura preoblicuje sve d, lj i nj u stringu s u ë (#235), æ (#156) i
å (#252) kad je smjer FALSE. Uz smjer TRUE i smjer je pretvorbe obrnut }
VAR
    i : BYTE;
BEGIN { DzLjNj }
    CASE smjer OF
        FALSE : BEGIN
            REPEAT
                i := POS('d', s);
                IF i > 0 THEN
                    BEGIN
                        DELETE(s, i, 1);
                        s[i] := 'ë';
                    END
                UNTIL i = 0;
            REPEAT
                i := POS('lj', s);
                IF i > 0 THEN
                    BEGIN
                        DELETE(s, i, 1);
                        s[i] := 'æ';
                    END
                UNTIL i = 0;
            REPEAT
                i := POS('nj', s);
                IF i > 0 THEN
                    BEGIN
                        DELETE(s, i, 1);
                        s[i] := 'å';
                    END
                UNTIL i = 0;
            END;
        TRUE : BEGIN
            REPEAT
                i := POS('ë', s);
                IF i > 0 THEN
                    BEGIN
                        s[i] := 'd';
                        INSERT(' ', s, i+1)
                    END
                UNTIL i = 0;
            REPEAT
                i := POS('æ', s);
                IF i > 0 THEN
                    BEGIN
                        s[i] := 'lj';
                        INSERT(' ', s, i+1)
                    END
                UNTIL i = 0;
            REPEAT
                i := POS('å', s);
                IF i > 0 THEN
                    BEGIN
                        s[i] := 'nj';
                        INSERT(' ', s, i+1)
                    END
                UNTIL i = 0;
            END;
        END
    END
END; { DzLjNj }

PROCEDURE UmetniA(VAR osn:str);
{ procedura umeće 'nepostojano' a ispred zadnjega pismena }
BEGIN { UmetniA }
    INSERT('a', osn, LENGTH(osn))
END; { UmetniA }

PROCEDURE Sibilizacija(VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Sibilizacija }
    CASE osn[i] OF
        'g' : osn[i] := 'z';
        'h' : osn[i] := 's';
        'k' : osn[i] := 'c';
    END
END; { Sibilizacija }

PROCEDURE Palatalizacija(VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Palatalizacija }
    CASE osn[i] OF
        'c' : osn[i] := 'č';
        'g' : osn[i] := 'đ';
        'h' : osn[i] := 't';
        'k' : osn[i] := 'š';
        'z' : osn[i] := 'ž';
    END
END; { Palatalizacija }

PROCEDURE Bipalatalizacija(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
    i : BYTE;
BEGIN { Bipalatalizacija }
    CASE osn[lnosn] OF
        'k' : IF ((osn[lnosn-2] = 's') AND (osn[lnosn-1] = 't')) OR
                ((osn[lnosn-2] = 'z') AND (osn[lnosn-1] = 'd')) THEN
            osn := COPY(osn, 1, lnosn-3) + '~';
    END
END; { Bipalatalizacija }

```



```

        'l','n' : CASE osn[lnosn-1] OF
                    's' : osn[lnosn-1] := '{';
                    'z' : osn[lnosn-1] := '{';
                END;
        '}', '~' : CASE osn[lnosn-1] OF
                    's' : osn[lnosn-1] := '{';
                    'z' : osn[lnosn-1] := '{';
                    'z' : osn[lnosn-1] := '{';
                END
            END
        END; { Bipalatalizacija }

PROCEDURE Jotacija(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { Jotacija }
    CASE osn[lnosn] OF
        'd' : osn[lnosn] := '|';
        's' : osn[lnosn] := '{';
        't' : osn[lnosn] := '{';
        'z' : osn[lnosn] := '{';
    END
END; { Jotacija }

PROCEDURE VokaliziranjeL(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { VokaliziranjeL }
    IF osn[lnosn-1]='l' THEN
        osn[lnosn-1]:='o'
    ELSE
        BEGIN
            IF osn[lnosn]='l' THEN
                osn[lnosn]:='o'
            END
        END
    END; { VokaliziranjeL }

PROCEDURE OdbaciKonacniC(VAR osn:str);
BEGIN { OdbaciKonacniC }
    DEC(osn[0])
END; { OdbaciKonacniC }

PROCEDURE Obezvucavanje(VAR osn:str);
VAR
    nadjeno : BOOLEAN;
    i : BYTE;
BEGIN { Obezvucavanje }
    nadjeno := FALSE;
    i := LENGTH(osn);
    REPEAT
        IF (osn[i-1] IN zvucni) AND (osn[i] IN bezvucni) THEN
            nadjeno := TRUE
        ELSE
            DEC(i)
        UNTIL nadjeno OR (i=1);
        IF nadjeno THEN
            CASE osn[i-1] OF
                'b' : osn[i-1] := '{';
                'd' : osn[i-1] := 'p';
                'g' : osn[i-1] := 't';
                'z' : osn[i-1] := 'k';
                'l' : osn[i-1] := 's';
                'l' : osn[i-1] := 's';
                'e' : osn[i-1] := '~';
            END
        END
    END; { Obezvucavanje }

PROCEDURE IzbaciUnutarnjiC(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { IzbaciUnutarnjiC }
    CASE osn[lnosn] OF
        'c' : IF osn[lnosn-1] IN ['d','t'] THEN
            DELETE(osn,lnosn-1,1);
        'k' : BEGIN
            IF osn[lnosn-1]='t' THEN
                CASE osn[lnosn-2] OF
                    's','{' : DELETE(osn,lnosn-1,1);
                END
            ELSE
                BEGIN
                    IF osn[lnosn-1]='d' THEN
                        CASE osn[lnosn-2] OF
                            'z','{' : BEGIN
                                DELETE(osn,lnosn-1,1);
                                Obezvucavanje(osn)
                            END;
                        END
                    END
                END
            END;
        'l' : IF (osn[lnosn-1]='t') AND (osn[lnosn-2]='s') THEN
            DELETE(osn,lnosn-1,1);
        'n' : CASE osn[lnosn-1] OF
            'd' : IF osn[lnosn-2]='' THEN
                DELETE(osn,lnosn-1,1);
            't' : IF osn[lnosn-2]='s' THEN
                DELETE(osn,lnosn-1,1)
            END
        END
    END
END; { IzbaciUnutarnjiC }

PROCEDURE ProsiriSJ(VAR osn:str);
BEGIN { ProsiriSJ }
    osn := osn + 'j'
END; { ProsiriSJ }

PROCEDURE KratiJatUI(VAR osn:str);
VAR
    i : BYTE;
BEGIN { KratiJatUI }
    i := POS('ije',osn);
    IF i>0 THEN
        DELETE(osn,i+1,2)
    ELSE
        BEGIN
            i := POS('je',osn);
            IF i>0 THEN
                BEGIN
                    DELETE(osn,i,1);
                    osn[i] := 'i';
                END
            END
        END
    END; { KratiJatUI }

PROCEDURE ProsiriSe(VAR osn:str);
BEGIN { ProsiriSe }
    osn := osn + 'er'
END; { ProsiriSe }

PROCEDURE StegniV(VAR osn:str);

```

```

VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniV }
  IF (osn[lnosn] IN samoglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN samoglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
END; { StegniV }

PROCEDURE StegniC(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniC }
  IF (osn[lnosn] IN suglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN suglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
END; { StegniC }

PROCEDURE ObezvacavanjeBezOkoline(VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i : BYTE;
BEGIN { ObezvacavanjeBezOkoline }
  nadjeno := FALSE;
  i := LENGTH(osn);
  REPEAT
    IF osn[i] IN zvucni THEN
      nadjeno := TRUE
    ELSE
      DEC(i)
  UNTIL nadjeno OR (i=1);
  IF nadjeno THEN
    CASE osn[i] OF
      'a' : osn[i] := ' ';
      'b' : osn[i] := 'p';
      'd' : osn[i] := 't';
      'g' : osn[i] := 'k';
      'z' : osn[i] := 's';
      'l' : osn[i] := 'r';
      'e' : osn[i] := '^';
    END
  END
END; { ObezvacavanjeBezOkoline }

PROCEDURE Umetnia2(VAR osn:str);
{ procedura umece 'nepostojano' a ispred predzadnjega pismena }
BEGIN { Umetnia }
  INSERT('a',osn,LENGTH(osn)-1)
END; { Umetnia }

PROCEDURE SrediJat(VAR osn:str);
VAR
  i : BYTE;
BEGIN { SrediJat }
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      DELETE(osn,i,1);
      IF (osn[i-1]='r') AND (osn[i-2] IN suglasni+zvukni) THEN
        INSERT('e',osn,i)
      ELSE
        INSERT('je',osn,i)
      END
    END
  END
END; { SrediJat }

FUNCTION PreoblikaOsn(osn:str; uzorak:INTEGER):str;
{ funkcija preoblicuje osnovu u skladu sa uputama pohranjenim
  u TRNSPRID.TXT }
VAR
  preoblike : str4;
  lnpreoblike : BYTE ABSOLUTE preoblike;
  i : BYTE;
BEGIN { PreoblikaOsn }
  DzLjNj(osn,FALSE);
  preoblike := t[uzorak];
  FOR x := 1 TO lnpreoblike DO
    CASE preoblike[x] OF
      '1' : Umetnia(osn);
      '2' : Sibilarizacija(osn);
      '3' : Palatalizacija(osn);
      '4' : Bipalatalizacija(osn);
      '5' : Jotacija(osn);
      '6' : VokaliziranjeL(osn);
      '7' : OdbaciKonacniC(osn);
      '8' : IzbaciUnutarnjiC(osn);
      '9' : ProsiriSJ(osn);
      'a' : Obezvacavanje(osn);
      'b' : KratijatUI(osn);
      'd' : ProsiriSEr(osn);
      'e' : StegniV(osn);
      'f' : StegniC(osn);
      'g' : ObezvacavanjeBezOkoline(osn);
      'h' : Umetnia2(osn);
      'i' : SrediJat(osn);
    END;
  DzLjNj(osn,TRUE);
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      osn[i] := 'i';
      INSERT('je',osn,i+1)
    END;
  PreoblikaOsn := osn
END; { PreoblikaOsn }

BEGIN { GenPrId }
  WRITEln(MEMAVAIL);
  InitUzorke;
  InitPreoblike;
  WRITEln(MEMAVAIL);
  InitVars;
  osnova := PARAMSTR(1);
  line := PARAMSTR(2);
  { *** Nadji supletivne osnove *** }
  IF osnova='dobr' THEN osnova := 'bol';
  IF osnova='zl' THEN osnova := 'gor';
  IF osnova='malen' THEN osnova := 'manj';
  IF osnova='velik' THEN osnova := 've';
  IF (osnova='dug') AND (line='432') THEN osnova := 'du';
  VAL(line,uzorak,z);
  posnova := PreoblikaOsn(osnova,uzorak-430);
  line := posnova + u[uzorak-430] + ' 462';
  line2 := 'naj' + posnova + u[uzorak-430] + ' 462';
  EXEC('c:\jezici\tp\dr\genprid.exe',line);
  EXEC('c:\jezici\tp\dr\genprid.exe',line2)

```

```
END. { GenPrid }
```

## 5.2.2.5. GENPRIL

```
{ $M $4000,0,0 }
PROGRAM GenPril;
{ Program generira preobliku osnovne priloga tvoreci komparativnu osnovu.
  U TRNSKOMP.TXT nalazi se popis transformacija osnova u svim uzorcima
}
USES
  DOS;

CONST
  maxuzoraka = 12;

TYPE
  fonemi = SET OF CHAR;
  str     = STRING;
  str3    = STRING[3];
  str4    = STRING[4];
  arr4    = ARRAY[1..maxuzoraka] OF str4;

VAR
  infile, outfile      : TEXT;
  zvucni, bezvucni, samoglasni, suglasni, zvonki      : fonemi;
  u, t                 : arr4;
  inime, outime, line, line2, rijec, osnova, posnova   : str;
  line3                : str3;
  line4, broj          : str4;
  ch                   : CHAR;
  ln1, ln2, ln3, ln4   : BYTE ABSOLUTE;
  uzorak, n, x, y, z   : INTEGER;

PROCEDURE InitVars;
VAR
  i, j, k : BYTE;
BEGIN { InitVars }
  zvucni := ['b','d','g','z','t','è'];
  bezvucni := ['c','h','k','p','s','t','j','~'];
  samoglasni := ['a','e','i','o','u'];
  zvonki := ['f','j','l','m','n','r','v','æ','n'];
  suglasni := zvucni + bezvucni;
  line := '';
  osnova := '';
  posnova := '';
  uzorak := 0;
END; { InitVars }

PROCEDURE InitUzorke;
{ Procedura otvara datnicu NASTKOMP.TXT s nastavcima za sve uzorke, učitava
  ih i smjesta u ARRAY u. Datnica NASTKOMP.TXT ima ovakvu strk:
  broj_uzorka:3, nastavak:3 }
BEGIN { InitUzorke }
  { *** Isprazni uzorke *** }
  FOR x := 1 TO maxuzoraka DO
    u[x] := '';

  { *** Ispuni uzorke nastavcima *** }
  ASSIGN(infile, 'NASTKOMP.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, line3);
      WHILE line3[ln13] = ' ' DO
        DEC(line3[0]);
      u[x] := line3;
      READLN(infile);
    END;
  CLOSE(infile);
END; { InitUzorke }

PROCEDURE InitPreoblike;
{ Procedura otvara datnicu TRNSKOMP.TXT s preoblikama za sve uzorke učitava
  ih i smjesta u ARRAY t. Datnica TRNSKOMP.TXT ima ovakvu strk:
  broj_uzorka:3, preoblike:4 }
BEGIN { InitPreoblike }
  { *** Isprazni preoblike *** }
  FOR x := 1 TO maxuzoraka DO
    t[x] := '';

  { *** Ispuni preoblike *** }
  ASSIGN(infile, 'TRNSKOMP.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, ch);
      READ(infile, line4);
      WHILE line4[ln14] = ' ' DO
        DEC(line4[0]);
      t[x] := line4;
      READLN(infile);
    END;
  CLOSE(infile);
END; { InitPreoblike }

PROCEDURE DzLjNj(VAR s:str; smjer:BOOLEAN);
{ procedura preoblicuje sve d, lj i nj u stringu s u è (#235), æ (#156) i
  ð (#252) kad je smjer FALSE. Uz smjer TRUE i smjer je pretvorbe obrnut }
VAR
  i : BYTE;
BEGIN { DzLjNj }
  CASE smjer OF
    FALSE : BEGIN
      REPEAT
        i := POS('d', s);
        IF i > 0 THEN
          BEGIN
            DELETE(s, i, 1);
            s[i] := 'è';
          END
        UNTIL i = 0;
      REPEAT
        i := POS('lj', s);
        IF i > 0 THEN
          BEGIN
```

```

        DELETE (s,i,1);
        s[i] := 'æ';
    END
UNTIL i=0;
REPEAT
    i := POS('nj',s);
    IF i>0 THEN
        BEGIN
            DELETE (s,i,1);
            s[i] := 'n';
        END
    UNTIL i=0
END;
TRUE : BEGIN
    REPEAT
        i := POS('e',s);
        IF i>0 THEN
            BEGIN
                s[i] := 'd';
                INSERT('.',s,i+1)
            END
        UNTIL i=0;
    REPEAT
        i := POS('æ',s);
        IF i>0 THEN
            BEGIN
                s[i] := 'l';
                INSERT('j',s,i+1)
            END
        UNTIL i=0;
    REPEAT
        i := POS('a',s);
        IF i>0 THEN
            BEGIN
                s[i] := 'n';
                INSERT('j',s,i+1)
            END
        UNTIL i=0
    END
END
END; { DzLjNj }

PROCEDURE UmetniA (VAR osn:str);
{ procedura umece 'nepostojano' a ispred zadnjega pismena }
BEGIN { UmetniA }
    INSERT('a',osn,LENGTH(osn))
END; { UmetniA }

PROCEDURE Sibilizacija (VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Sibilizacija }
    CASE osn[i] OF
        'g' : osn[i] := 'z';
        'h' : osn[i] := 's';
        'k' : osn[i] := 'c';
    END
END; { Sibilizacija }

PROCEDURE Palatalizacija (VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Palatalizacija }
    CASE osn[i] OF
        'c' : osn[i] := '~';
        'd' : osn[i] := 'j';
        'g' : osn[i] := 'j';
        'k' : osn[i] := '~';
        'z' : osn[i] := '~';
    END
END; { Palatalizacija }

PROCEDURE Bipalatalizacija (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
    i : BYTE;
BEGIN { Bipalatalizacija }
    CASE osn[lnosn] OF
        'k' : IF ((osn[lnosn-2]='s') AND (osn[lnosn-1]='t')) OR
            ((osn[lnosn-2]='z') AND (osn[lnosn-1]='d')) THEN
            osn := COPY(osn,1,lnosn-3)+'{~';
        'l','n' : CASE osn[lnosn-1] OF
            's' : osn[lnosn-1] := '{';
            'z' : osn[lnosn-1] := '{';
        END;
        '}', '~' : CASE osn[lnosn-1] OF
            's' : osn[lnosn-1] := '{';
            'z' : osn[lnosn-1] := '{';
            '}' : osn[lnosn-1] := '{';
        END
    END
END; { Bipalatalizacija }

PROCEDURE Jotacija (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { Jotacija }
    CASE osn[lnosn] OF
        'd' : osn[lnosn] := '{';
        'g' : osn[lnosn] := '{';
        't' : osn[lnosn] := '{';
        'z' : osn[lnosn] := '{';
    END
END; { Jotacija }

PROCEDURE VokaliziranjeL (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { VokaliziranjeL }
    IF osn[lnosn-1]='l' THEN
        osn[lnosn-1]:='o'
    ELSE
        BEGIN
            IF osn[lnosn]='l' THEN
                osn[lnosn]:='o'
            END
        END
    END; { VokaliziranjeL }

PROCEDURE OdbaciKonacniC (VAR osn:str);
BEGIN { OdbaciKonacniC }
    DEC(osn[0])
END; { OdbaciKonacniC }

PROCEDURE Obezvucavanje (VAR osn:str);
VAR
    nadjeno : BOOLEAN;
    i : BYTE;
BEGIN { Obezvucavanje }
    nadjeno := FALSE;
    i := LENGTH(osn);
    REPEAT

```

```

        IF (osn[i-1] IN zvucni) AND (osn[i] IN bezvucni) THEN
            nadjeno := TRUE
        ELSE
            DEC(i)
        UNTIL nadjeno OR (i=1);
        IF nadjeno THEN
            CASE osn[i-1] OF
                ' ' : osn[i-1] := '{';
                'b' : osn[i-1] := 'p';
                'd' : osn[i-1] := 't';
                'g' : osn[i-1] := 'k';
                'z' : osn[i-1] := 's';
                '|' : osn[i-1] := '}';
                'è' : osn[i-1] := '~';
            END
        END; { Obezvucavanje }

PROCEDURE IzbaciUnutarnjiC (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { IzbaciUnutarnjiC }
    CASE osn[lnosn] OF
        'c' : IF osn[lnosn-1] IN ['d','t'] THEN
            DELETE(osn,lnosn-1,1);
        'k' : BEGIN
            IF osn[lnosn-1]='t' THEN
                CASE osn[lnosn-2] OF
                    's','{' : DELETE(osn,lnosn-1,1);
                END
            ELSE
                BEGIN
                    IF osn[lnosn-1]='d' THEN
                        CASE osn[lnosn-2] OF
                            'z','{' : BEGIN
                                DELETE(osn,lnosn-1,1);
                                Obezvucavanje(osn)
                            END;
                        END
                    END
                END
            END;
        'l' : IF (osn[lnosn-1]='t') AND (osn[lnosn-2]='s') THEN
            DELETE(osn,lnosn-1,1);
        'n' : CASE osn[lnosn-1] OF
            'd' : IF osn[lnosn-2]=' ' THEN
                DELETE(osn,lnosn-1,1);
            't' : IF osn[lnosn-2]='s' THEN
                DELETE(osn,lnosn-1,1)
            END
        END
    END
END; { IzbaciUnutarnjiC }

PROCEDURE ProsiriSJ (VAR osn:str);
BEGIN { ProsiriSJ }
    osn := osn + 'j'
END; { ProsiriSJ }

PROCEDURE KratiJatUI (VAR osn:str);
VAR
    i : BYTE;
BEGIN { KratiJatUI }
    i := POS('ije',osn);
    IF i>0 THEN
        DELETE(osn,i+1,2)
    ELSE
        BEGIN
            i := POS('je',osn);
            IF i>0 THEN
                BEGIN
                    DELETE(osn,i,1);
                    osn[i] := 'i';
                END
            END
        END
    END; { KratiJatUI }

PROCEDURE ProsiriSEr (VAR osn:str);
BEGIN { Prosiri s -er- }
    osn := osn + 'er'
END; { Prosiri s -er- }

PROCEDURE StegniV (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniV }
    IF (osn[lnosn] IN samoglasi) AND (osn[lnosn-1]=osn[lnosn]) THEN
        DEC(osn[0])
    ELSE
        BEGIN
            IF (osn[lnosn-1] IN samoglasi) AND (osn[lnosn-2]=osn[lnosn-1])
            THEN DELETE(osn,lnosn-1,1)
            END
        END
    END; { StegniV }

PROCEDURE StegniC (VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniC }
    IF (osn[lnosn] IN suglasi) AND (osn[lnosn-1]=osn[lnosn]) THEN
        DEC(osn[0])
    ELSE
        BEGIN
            IF (osn[lnosn-1] IN suglasi) AND (osn[lnosn-2]=osn[lnosn-1])
            THEN DELETE(osn,lnosn-1,1)
            END
        END
    END; { StegniC }

PROCEDURE ObezvucavanjeBezOkoline (VAR osn:str);
VAR
    nadjeno : BOOLEAN;
    i : BYTE;
BEGIN { ObezvucavanjeBezOkoline }
    nadjeno := FALSE;
    i := LENGTH(osn);
    REPEAT
        IF osn[i] IN zvucni THEN
            nadjeno := TRUE
        ELSE
            DEC(i)
        UNTIL nadjeno OR (i=1);
        IF nadjeno THEN
            CASE osn[i] OF
                ' ' : osn[i] := '{';
                'b' : osn[i] := 'p';
                'd' : osn[i] := 't';
                'g' : osn[i] := 'k';
                'z' : osn[i] := 's';
                '|' : osn[i] := '}';
                'è' : osn[i] := '~';
            END
        END
    END; { ObezvucavanjeBezOkoline }

PROCEDURE UmetniA2 (VAR osn:str);
{ procedura umece 'nepostojano' a ispred predzadnjega pismena }

```

```

BEGIN { UmetniA }
  INSERT('a',osn,LENGTH(osn)-1)
END; { UmetniA }

PROCEDURE SrediJat(VAR osn:str);
VAR
  i : BYTE;
BEGIN { SrediJat }
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      DELETE(osn,i,1);
      IF (osn[i-1]='r') AND (osn[i-2] IN suglasi+zvonki) THEN
        INSERT('e',osn,i)
      ELSE
        INSERT('je',osn,i)
      END
    END
  END; { SrediJat }

FUNCTION PreoblikaOsn(osn:str; uzorak:INTEGER):str;
{ funkcija preoblicuje osnovu u skladu sa uputama pohranjenim
u TRNSPRID.TXT}
VAR
  preoblike : str4;
  lnpreoblike : BYTE ABSOLUTE preoblike;
  i : BYTE;
BEGIN { PreoblikaOsn }
  DzLjNj(osn,FALSE);
  preoblike := t[uzorak];
  FOR x := 1 TO lnpreoblike DO
    CASE preoblike[x] OF
      '1' : UmetniA(osn);
      '2' : Sibilizacija(osn);
      '3' : Palatalizacija(osn);
      '4' : Bipalatalizacija(osn);
      '5' : Jotacija(osn);
      '6' : Vokaliziranje(osn);
      '7' : OdbacivanjeC(osn);
      '8' : IzbacivanjeC(osn);
      '9' : ProsiranjeC(osn);
      'a' : Obezvučavanje(osn);
      'b' : KratiranjeC(osn);
      'd' : ProsiranjeS(osn);
      'e' : Stegnutost(osn);
      'f' : StegnutostC(osn);
      'g' : ObezvučavanjeBezOkoline(osn);
      'h' : UmetniA2(osn);
      'i' : SrediJat(osn);
    END;
  DzLjNj(osn,TRUE);
  i := POS('e',osn);
  IF i>0 THEN
    BEGIN
      osn[i] := 'i';
      INSERT('je',osn,i+1)
    END;
  PreoblikaOsn := osn
END; { PreoblikaOsn }

BEGIN { GenPrid }
  WRITELN(MEMAVAIL);
  InitUzorke;
  InitPreoblike;
  WRITELN(MEMAVAIL);
  InitVars;
  rijec := PARAMSTR(1);
  WHILE rijec[lnr] IN ['1'..'9'] DO
    DEC(lnr);
    osnova := PARAMSTR(2);
    line := PARAMSTR(3);

    { *** Nadji supletivne osnove *** }
    IF osnova='dobr' THEN osnova := 'bol';
    IF osnova='zl' THEN osnova := 'gor';
    IF osnova='malen' THEN osnova := 'manj';
    IF osnova='velik' THEN osnova := 've';
    IF (osnova='dug') AND (line='432') THEN osnova := 'du';

    VAL(line,uzorak,z);
    posnova := PreoblikaOsn(osnova,uzorak-430);
    WRITELN(rijec:20,posnova+u[uzorak-430]+'e':20,'naj'+posnova+u[uzorak-430]+'e':20)
  END. { GenPrid }

```

## 5.2.2.6. GENGLAG

```

PROGRAM GenGlag;
{ Program generira oblike glagola. U NASTGLAG.TXT nalaze se nastavci.
  U TRNSGLAG.TXT nalazi se popis transformacija osnova u svim uzorcima
}
USES
  DOS;

CONST
  maxuzoraka = 153;

TYPE
  fonemi = SET OF CHAR;
  str = STRING;
  str3 = STRING[3];
  str4 = STRING[4];
  str5 = STRING[5];
  str20 = STRING[18];
  ut = ARRAY[1..maxuzoraka,1..9,1..6] OF str4;
  utptr = ^ut;
  joblici = ARRAY[1..9,1..6] OF str4;

VAR
  zvucni, bezvucni, samoglasni, suglasni, zvonki : fonemi;
  u, t : utptr;
  prdgms : ARRAY[1..9,1..6] OF str20;
  svrs : ARRAY[1..maxuzoraka] OF CHAR;
  osn : str;
  line, rijec : str20;
  line3 : str3;
  line5, broj : str5;
  ch : CHAR;
  lnr : BYTE ABSOLUTE rijec;
  ln13 : BYTE ABSOLUTE line3;
  ln15 : BYTE ABSOLUTE line5;
  uzorak, n, x, y, z : INTEGER;
  reflex : BOOLEAN;

PROCEDURE InitVars;
VAR

```

```

i, j, k : BYTE;
BEGIN { InitVars }
  zvucni := ['\','b','d','g','z','\','è'];
  bezvucni := ['c','h','k','p','s','t','\','\','\','\'];
  samoglasni := ['a','e','i','o','u'];
  zvonki := ['f','j','l','m','n','r','v','æ','n'];
  suglasni := zvucni + bezvucni;
  line := '';
  osn := '';
  uzorak := 0;
  reflex := FALSE;
  FOR i := 1 TO 9 DO
    FOR j := 1 TO 6 DO
      prdgms[i,j] := ' '
    END
  END; { InitVars }

PROCEDURE InitUzorke;
{ Procedura otvara datnicu NASTKOMP.TXT s nastavcima za sve uzorke, učitava
ih i smjesta u ARRAY u. Datnica NASTKOMP.TXT ima ovakvu strk:
broj_uzorka:3, nastavak:4 }
VAR
  infile : TEXT;
  inime : str;

BEGIN { InitUzorke }
  NEW(u);
  { *** Isprazni uzorke *** }
  FOR x := 1 TO maxuzoraka DO
    FOR y := 1 TO 9 DO
      FOR z := 1 TO 6 DO
        u^[x,y,z] := ' ';
      END
    END
  END;
  { *** Ispuni uzorke nastavcima *** }
  ASSIGN(infile,'NASTGLAG.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      FOR y := 1 TO 9 DO
        FOR z := 1 TO 6 DO
          BEGIN
            READ(infile,line5);
            WHILE line5[lnl5]=' ' DO
              DEC(line5[0]);
            u^[x,y,z] := line5;
          END;
        END;
      READLN(infile,svrs[x])
    END;
  END;
  CLOSE(infile)
END; { InitUzorke }

PROCEDURE InitPreoblike;
{ Procedura otvara datnicu TRNSKOMP.TXT s preoblikama za sve uzorke učitava
ih i smjesta u ARRAY t. Datnica TRNSKOMP.TXT ima ovakvu strk:
broj_uzorka:3, preoblike:4 }
VAR
  infile : TEXT;
  inime : str;

BEGIN { InitPreoblike }
  NEW(t);
  { *** Isprazni preoblike *** }
  FOR x := 1 TO maxuzoraka DO
    FOR y := 1 TO 9 DO
      FOR z := 1 TO 6 DO
        t^[x,y,z] := ' ';
      END
    END
  END;
  { *** Ispuni preoblike *** }
  ASSIGN(infile,'TRNSGLAG.TXT'); RESET(infile);
  FOR x := 1 TO maxuzoraka DO
    BEGIN
      READ(infile,ch);
      READ(infile,ch);
      READ(infile,ch);
      FOR y := 1 TO 9 DO
        FOR z := 1 TO 6 DO
          BEGIN
            READ(infile,line5);
            WHILE line5[lnl5]=' ' DO
              DEC(line5[0]);
            t^[x,y,z] := line5;
          END;
        END;
      READLN(infile)
    END;
  END;
  CLOSE(infile)
END; { InitPreoblike }

PROCEDURE DzLjNj(VAR s:str; smjer:BOOLEAN);
{ procedura preoblicuje sve d', lj i nj u stringu s u è (#235), æ (#156) i
u (#252) kad je smjer FALSE. Uz smjer TRUE i smjer je pretvorbe obrnut }
VAR
  i : BYTE;

BEGIN { DzLjNj }
  CASE smjer OF
    FALSE : BEGIN
      REPEAT
        i := POS('d',s);
        IF i>0 THEN
          BEGIN
            DELETE(s,i,1);
            s[i] := 'è';
          END
        UNTIL i=0;
      REPEAT
        i := POS('lj',s);
        IF i>0 THEN
          BEGIN
            DELETE(s,i,1);
            s[i] := 'æ';
          END
        UNTIL i=0;
      REPEAT
        i := POS('nj',s);
        IF i>0 THEN
          BEGIN
            DELETE(s,i,1);
            s[i] := 'u';
          END
        UNTIL i=0;
      END;
    TRUE : BEGIN
      REPEAT
        i := POS('è',s);
        IF i>0 THEN
          BEGIN
            s[i] := 'd';
            INSERT(' ',s,i+1)
          END
        UNTIL i=0;
      REPEAT
        i := POS('æ',s);
        IF i>0 THEN

```

```

        BEGIN
            s[i] := 'l';
            INSERT('j',s,i+1)
        END
    UNTIL i=0;
REPEAT
    i := POS('a',s);
    IF i>0 THEN
        BEGIN
            s[i] := 'n';
            INSERT('j',s,i+1)
        END
    END
UNTIL i=0
END
END
END; { DzLjNj }

PROCEDURE Umetnia(VAR osn:str);
{ procedura umece 'nepostojano' a ispred zadnjega pismena }
BEGIN { Umetnia }
    INSERT('a',osn,LENGTH(osn))
END; { Umetnia }

PROCEDURE Sibilizacija(VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Sibilizacija }
    CASE osn[i-1] OF
        'g' : osn[i-1] := 'z';
        'h' : osn[i-1] := 's';
        'k' : osn[i-1] := 'c';
    ELSE
        CASE osn[i] OF
            'g' : osn[i] := 'z';
            'h' : osn[i] := 's';
            'k' : osn[i] := 'c';
        END
    END
END; { Sibilizacija }

PROCEDURE Palatalizacija(VAR osn:str);
VAR
    i : BYTE ABSOLUTE osn;
BEGIN { Palatalizacija }
    CASE osn[i] OF
        'c' : osn[i] := '~';
        'g' : osn[i] := '~';
        'h' : osn[i] := '~';
        'k' : osn[i] := '~';
        'z' : osn[i] := '~';
    END
END; { Palatalizacija }

PROCEDURE Bipalatalizacija(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
    i : BYTE;
BEGIN { Bipalatalizacija }
    CASE osn[lnosn-1] OF
        'l','n' : CASE osn[lnosn-2] OF
            's' : osn[lnosn-2] := '{';
            'z' : osn[lnosn-2] := '{';
        END
    ELSE
        CASE osn[lnosn] OF
            'k' : IF ((osn[lnosn-2]='s') AND (osn[lnosn-1]='t')) OR
                ((osn[lnosn-2]='z') AND (osn[lnosn-1]='d')) THEN
                osn := COPY(osn,1,lnosn-3)+'~';
            'l','n' : CASE osn[lnosn-1] OF
                's' : osn[lnosn-1] := '{';
                'z' : osn[lnosn-1] := '{';
            END;
            '}', '~' : CASE osn[lnosn-1] OF
                's' : osn[lnosn-1] := '{';
                'z' : osn[lnosn-1] := '{';
                '~' : osn[lnosn-1] := '{';
            END;
            'a','e','ë','i' : IF (osn[lnosn-2] IN {'s','z'}) AND
                (osn[lnosn-1] IN {'d','t'}) THEN
                CASE osn[lnosn-2] OF
                    's' : osn[lnosn-2] := '{';
                    'z' : osn[lnosn-2] := '{';
                END
            END
    END
END; { Bipalatalizacija }

PROCEDURE Jotacija(VAR os:str);
VAR
    lnos : BYTE ABSOLUTE os;
BEGIN { Jotacija }
    CASE os[lnos-1] OF
        'l','p','v','f','m' : INSERT('lj',os,lnos);
        'l','n' : INSERT('lj',os,lnos);
        's','h' : os[lnos-1] := '{';
        'g','z' : os[lnos-1] := '{';
        't' : os[lnos-1] := '{';
        'd' : os[lnos-1] := '{';
        'c','k' : os[lnos-1] := '~';
    ELSE
        BEGIN
            CASE os[lnos] OF
                'b','p','v','f','m' : os := os + 'lj';
                'l','n' : os := os + 'lj';
                's','h' : os[lnos] := '{';
                'g','z' : os[lnos] := '{';
                't' : os[lnos] := '{';
                'd' : os[lnos] := '{';
                'c','k' : os[lnos] := '~';
            END
        END
    END
END; { Jotacija }

PROCEDURE VokaliziranjeL(VAR osn:str);
VAR
    lnosn : BYTE ABSOLUTE osn;
BEGIN { VokaliziranjeL }
    IF osn[lnosn-1]='l' THEN
        osn[lnosn-1]:='o'
    ELSE
        BEGIN
            IF osn[lnosn]='l' THEN
                osn[lnosn]:='o'
            END
        END
    END; { VokaliziranjeL }

PROCEDURE OdbaciKonacniC(VAR osn:str);
BEGIN { OdbaciKonacniC }
    DEC(osn[0])
END; { OdbaciKonacniC }

```



```

PROCEDURE Obezvucavanje(VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i       : BYTE;
BEGIN { Obezvucavanje }
  nadjeno := FALSE;
  i := LENGTH(osn);
  REPEAT
    IF (osn[i-1] IN zvucni) AND (osn[i] IN bezvucni) THEN
      nadjeno := TRUE
    ELSE
      DEC(i)
    UNTIL nadjeno OR (i=1);
    IF nadjeno THEN
      CASE osn[i-1] OF
        ' ' : osn[i-1] := ' ';
        'b' : osn[i-1] := 'p';
        'd' : osn[i-1] := 't';
        'g' : osn[i-1] := 'k';
        'z' : osn[i-1] := 's';
        'l' : osn[i-1] := ' ';
        'e' : osn[i-1] := '~';
      END
    END
  END; { Obezvucavanje }

PROCEDURE IzbaciUnutarnjiC(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { IzbaciUnutarnjiC }
  CASE osn[lnosn] OF
    'c' : IF osn[lnosn-1] IN ['d','t'] THEN
      DELETE(osn,lnosn-1,1);
    'k' : BEGIN
      IF osn[lnosn-1]='t' THEN
        CASE osn[lnosn-2] OF
          's','{' : DELETE(osn,lnosn-1,1);
        END
      ELSE
        BEGIN
          IF osn[lnosn-1]='d' THEN
            CASE osn[lnosn-2] OF
              'z','{' : BEGIN
                DELETE(osn,lnosn-1,1);
                Obezvucavanje(osn);
              END;
            END
          END;
        END;
      END;
    'l' : IF (osn[lnosn-1]='t') AND (osn[lnosn-2]='s') THEN
      DELETE(osn,lnosn-1,1);
    'n' : CASE osn[lnosn-1] OF
      'd' : IF osn[lnosn-2]=' ' THEN
        DELETE(osn,lnosn-1,1);
      't' : IF osn[lnosn-2]='s' THEN
        DELETE(osn,lnosn-1,1);
      END;
    'e' : IF (osn[lnosn-3]='s') AND
      (osn[lnosn-2]='t') AND
      (osn[lnosn-1]='k') THEN
      DELETE(osn,lnosn-2,1)
    END
  END
END; { IzbaciUnutarnjiC }

PROCEDURE ProsiriSJ(VAR osn:str);
BEGIN { ProsiriSJ }
  osn := osn + 'j'
END; { ProsiriSJ }

PROCEDURE JatUU(VAR osn:str);
VAR
  i : BYTE;
BEGIN { JatUU }
  i := POS(' ',osn);
  IF i>0 THEN
    osn[i] := 'u'
  END; { JatUU }

PROCEDURE KratiJatUI(VAR osn:str);
VAR
  i : BYTE;
BEGIN { KratiJatUI }
  i := POS('ije',osn);
  IF i>0 THEN
    DELETE(osn,i+1,2)
  ELSE
    BEGIN
      i := POS('je',osn);
      IF i>0 THEN
        BEGIN
          DELETE(osn,i,1);
          osn[i] := 'i';
        END
      END
    END
  END; { KratiJatUI }

PROCEDURE ProsiriSEr(VAR osn:str);
BEGIN { Prosiri s -er- }
  osn := osn + 'er'
END; { Prosiri s -er- }

PROCEDURE StegniV(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniV }
  IF (osn[lnosn] IN samoglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN samoglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
  END; { StegniV }

PROCEDURE StegniC(VAR osn:str);
VAR
  lnosn : BYTE ABSOLUTE osn;
BEGIN { StegniC }
  IF (osn[lnosn] IN suglasni) AND (osn[lnosn-1]=osn[lnosn]) THEN
    DEC(osn[0])
  ELSE
    BEGIN
      IF (osn[lnosn-1] IN suglasni) AND (osn[lnosn-2]=osn[lnosn-1])
        THEN DELETE(osn,lnosn-1,1)
      END
    END
  END; { StegniC }

PROCEDURE ObezvucavanjeBezOkoline(VAR osn:str);
VAR
  nadjeno : BOOLEAN;
  i       : BYTE;
BEGIN { ObezvucavanjeBezOkoline }

```

```

nadjeno := FALSE;
i := LENGTH(osn);
REPEAT
    IF osn[i] IN zvucni THEN
        nadjeno := TRUE
    ELSE
        DEC(i)
    UNTIL nadjeno OR (i=1);
IF nadjeno THEN
    CASE osn[i] OF
        'c' : osn[i] := 'č';
        'b' : osn[i] := 'p';
        'd' : osn[i] := 't';
        'g' : osn[i] := 'k';
        'z' : osn[i] := 's';
        'l' : osn[i] := 'j';
        'e' : osn[i] := 'š';
    END
END; { ObezvucavanjeBezOkoline }

PROCEDURE UmetniA2(VAR osn:str);
{ procedura umeće 'nepostojano' a ispred predzadnjega pismena }
BEGIN { UmetniA }
    INSERT('a',osn,LENGTH(osn)-1)
END; { UmetniA }

PROCEDURE SrediJat(VAR osn:str);
VAR
    i : BYTE;
BEGIN { SrediJat }
    i := POS('ě',osn);
    IF i>0 THEN
        BEGIN
            DELETE(osn,i,1);
            IF (osn[i-1]='r') AND (osn[i-2] IN suglasizvsonki) THEN
                INSERT('e',osn,i)
            ELSE
                INSERT('je',osn,i)
            END
        END
    END; { SrediJat }

PROCEDURE EuO(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { EuO }
    IF os[o]='e' THEN
        BEGIN
            DEC(os[0]);
            os := os + 'o'
        END
    END; { EuO }

PROCEDURE OdbaciPretkonacniC(VAR os:str);
{ procedura izbacuje pretkonacni C }
BEGIN { OdbaciPretkonacniC }
    DELETE(os,LENGTH(os)-1,1)
END; { OdbaciPretkonacniC }

PROCEDURE EuA(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { EuA }
    IF os[o]='e' THEN
        BEGIN
            DEC(os[0]);
            os := os + 'a'
        END
    END; { EuA }

PROCEDURE EuI(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { EuI }
    IF os[o]='e' THEN
        BEGIN
            DEC(os[0]);
            os := os + 'i'
        END
    END; { EuI }

PROCEDURE EuU(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { EuU }
    IF os[o]='e' THEN
        BEGIN
            DEC(os[0]);
            os := os + 'u'
        END
    END; { EuU }

PROCEDURE JatuI(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { JatuI }
    IF os[o]='ě' THEN
        os[o] := 'i'
    ELSE
        BEGIN
            IF os[o-1]='ě' THEN
                os[o-1] := 'i'
            ELSE
                BEGIN
                    IF os[o-2]='ě' THEN
                        os[o-2] := 'i'
                    END
                END
            END
        END
    END; { JatuI }

PROCEDURE JatuE(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;
BEGIN { JatuE }
    IF os[o]='ě' THEN
        os[o] := 'e'
    ELSE
        BEGIN
            IF os[o-1]='ě' THEN
                os[o-1] := 'e'
            ELSE
                BEGIN
                    IF os[o-2]='ě' THEN
                        os[o-2] := 'e'
                    END
                END
            END
        END
    END; { JatuE }

PROCEDURE IuE(VAR os:str);
VAR
    o : BYTE ABSOLUTE os;

```

```

BEGIN { IuE }
  IF os[o]='i' THEN
    BEGIN
      DEC(os[0]);
      os := os + 'e'
    END
  END; { IuE }

PROCEDURE OuA(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { OuA }
  IF os[o-2]='o' THEN
    os[o-2] := 'a'
  END; { OuA }

PROCEDURE VEuUjE(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { VEuUjE }
  IF (os[o]='e') AND (os[o-1] IN ['e','i','o']) THEN
    BEGIN
      os[o-1] := 'u';
      INSERT('j',os,o)
    END
  END; { VEuUjE }

PROCEDURE UmetniV(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { UmetniV }
  INSERT('v',os,o)
END; { UmetniV }

PROCEDURE IzbaciPretpretkonacniC(VAR os:str);
BEGIN { OdbaciPretpretkonacniC }
  DELETE(os,LENGTH(os)-2,1)
END; { OdbaciPretpretkonacniC }

PROCEDURE VuJ(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { VuJ }
  IF os[o-1]='v' THEN
    os[o-1] := 'j'
  END; { VuJ }

PROCEDURE UmetniUv(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { UmetniUv }
  INSERT('uv',os,o)
END; { UmetniUv }

PROCEDURE DodajVe(VAR os:str);
VAR
  o : BYTE ABSOLUTE os;
BEGIN { DodajVe }
  os := os + 've'
END; { DodajVe }

FUNCTION PreoblikaOsn(osnv:str; uzorak:INTEGER; v1, l1:BYTE):str;
{ funkcija preoblicuje osnovu u skladu sa uputama pohranjenim
  u TRNSPRID.TXT}
VAR
  preoblike : str5;
  lnpreoblike : BYTE ABSOLUTE preoblike;
  l : BYTE;
BEGIN { PreoblikaOsn }
  DzLjNj(osnv,FALSE);
  preoblike := t'[uzorak,v1,l1];
  FOR x := 1 TO lnpreoblike DO
    CASE preoblike[x] OF
      '1' : UmetniA(osnv);
      '2' : Sibilizacija(osnv);
      '3' : Palatalizacija(osnv);
      '4' : Bipalatalizacija(osnv);
      '5' : Jotacija(osnv);
      '6' : Vokaliziranje(osnv);
      '7' : OdbaciKonacniC(osnv);
      '8' : IzbaciUnutarnjiC(osnv);
      '9' : ProsiriSJ(osnv);
      'a' : Obezvacavanje(osnv);
      'b' : KratiJatUI(osnv);
      'c' : JatUI(osnv);
      'd' : ProsiriSEr(osnv);
      'e' : StegniV(osnv);
      'f' : StegniC(osnv);
      'g' : ObezvacavanjeBezOkoline(osnv);
      'h' : UmetniH(osnv);
      'i' : SrediJat(osnv);
      'j' : EuO(osnv);
      'm' : OdbaciPretkonacniC(osnv);
      'n' : EuA(osnv);
      'o' : EuI(osnv);
      'p' : EuU(osnv);
      'q' : JatUI(osnv);
      'r' : JatUE(osnv);
      's' : IuE(osnv);
      't' : OuA(osnv);
      'u' : VEuUjE(osnv);
      'v' : UmetniV(osnv);
      'w' : IzbaciPretpretkonacniC(osnv);
      'x' : VuJ(osnv);
      'y' : UmetniUv(osnv);
      'z' : DodajVe(osnv);
    END;
  DzLjNj(osnv,TRUE);
  i := POS('e',osnv);
  IF i>0 THEN
    BEGIN
      osnv[i] := 'i';
      INSERT('je',osnv,i+1)
    END;
  PreoblikaOsn := osnv
END; { PreoblikaOsn }

PROCEDURE GenParadigme(VAR osnova:str; uzorak:INTEGER);
VAR
  posnova : str;
  v,l : BYTE;
BEGIN { GenParadigme }
  FOR v := 1 TO 9 DO
    FOR l := 1 TO 6 DO
      BEGIN
        rijec := u^[uzorak,v,l];
        IF rijec<>'.' THEN
          BEGIN

```

```

                                posnova := PreoblikaOsn(osnova,uzorak,v,l);
                                rijec := posnova+rijec;
                                IF reflex AND (v IN [1,2,3,4,5,8,9]) THEN
                                    rijec := rijec+' se';
                                END;
                                prdgms[v,l] := rijec
                            END
                        END; { GenParadigme }
END; { GenParadigme }

PROCEDURE PisiParadigme;
BEGIN { PisiParadigme }
    WRITE(uzorak);
    CASE svrs[uzorak-500] OF
        '0' : Writeln(' nesvr{.}');
        '1' : Writeln(' svr{.}');
        '2' : Writeln(' nesvr{.} & svr{.}')
    END;
    FOR x := 1 TO 6 DO
        Writeln(prdgms[1,x]:20,prdgms[2,x]:20,prdgms[3,x]:20);
    Writeln;
    FOR x := 1 TO 6 DO
        Writeln(prdgms[4,x]:20,prdgms[5,x]:20,prdgms[6,x]:20);
    Writeln;
    FOR x := 1 TO 6 DO
        Writeln(prdgms[7,x]:20,prdgms[8,x]:20,prdgms[9,x]:20);
    Writeln
END; { PisiParadigme }

BEGIN { GenGlag }
    Writeln(MEMAVAIL);
    InitUzorke;
    InitPreoblike;
    Writeln(MEMAVAIL);
    InitVars;
    rijec := PARAMSTR(1);
    osn := PARAMSTR(2);
    line := PARAMSTR(3);
    {osn := 'mel6';
    line := '613';}
    IF POS(' se',rijec)<>0 THEN
        reflex := TRUE;
    VAL(line,uzorak,z);

    { *** dodaj osnovi docetne samoglasne *** }
    CASE uzorak OF
        501..502 : osn := osn + 'a';
        503..511 : osn := osn + 'i';
        512..514 : osn := osn + 'e';
        515..587 : osn := osn + 'e';
        588..596 : osn := osn + 'e';
        597..612 : osn := osn + 'e';
        613..621 : osn := osn + 'e';
        622..622 : osn := osn + 'e';
        623..623 : osn := osn + 'e';
        624..647 : osn := osn + 'e';
        648..651 : osn := osn + 'e';
        652..653 : osn := osn + 'e';
    END;
    GenParadigme(osn,uzorak-500);
    PisiParadigme
END. { GenGlag }

```

## 6. ZAKLJUČAK I PERSPEKTIVE

Cilj je ovoga rada bio pokušati opisati načine promjene oblika riječi u hrvatskome. Taj opis svoje teorijske oslonce nalazi u dva lingvistička područja.

Prvo je onaj dio naslijeđa generativne gramatike koji se nazivlje generativna fonologija tj. onaj njezin dio zadužen za obradu flektivne morfologije. Metodološki koncepti koji su iz njega preuzeti sastoje se od dvije razine reprezentacije — morfotaktičke (ulazne) i preobličene (izlazne) te preoblikâ kojima su one povezane. Ono što se, međutim, od generativne fonologije ne preuzimlje, koncepcija je visokoformaliziranih morfološki i fonološki uvjetovanih alternacija tj. preoblika koje se primjenjuju na globalnoj razini. Upravo usuprot tome, model GENOBLIK zastupa tezu, koju izvedbom i pokazuje, da se alternacije, koje se pojavljuju u fleksiji hrvatskoga, mogu opisati precizno određenim kontekstima vezanim uz uzorke promjene oblika. Teorijsko opravdanje za uključivanje uzorka promjene u globalni sustav generativne gramatike može se naći u Chomsky (1965:171-172) gdje se upućuje i na drugo teorijsko područje-oslonac ovoga rada.

Riječ je o tradicionalnogramatičkoj klasifikaciji uzoraka promjene, koja je klasifikacija postala temeljem za novu, osjetljiviju na dvije razine reprezentacije i alternacije te na razne morfološke obavijesti. Cilj je te nove klasifikacije bio opisati načine promjene oblika riječi u hrvatskome. To je izvedeno uz postuliranje dvaju konstrukata: *vrste promjene* (tri osnovna tipa: deklinacija, konjugacija, komparacija) i *uzorka promjene*. Uzorkom se, struktura kojeg je određena paradigmom karakterističnom za vrstu promjene, određuju nastavci što se osnovi moraju pridružiti i preobliske

koje se nad njom moraju obaviti ne bi li se kao konačni rezultat dobio ovjereni oblik riječi. Uzorci su promjene predmet klasifikacije, a kriteriji koji su korišteni pri klasificiranju specifični su za svaku vrstu promjene.

Kako je ovdje riječ o klasifikaciji koja vodi računa samo o načinima promjene oblika, u daljnjim bi istraživanjima bilo zanimljivo ispitati kakva je korespondencija oblika riječi iz tako postavljenih vrsta i uzoraka promjene s njihovim sintaktičkim funkcijama. Dakako, da bi u hrvatskome i inače valjalo provesti klasifikaciju vrsta riječi s obzirom na njihovu službu u rečenici tj. njihovu sintaktičku funkciju. Zacijelo bi najzanimljiviji rezultati izašli iz sraza takve klasifikacije s ovom upravo danom.

Tako postavljen opis flektivne morfologije hrvatskoga pokazao se pogodnim za izvedbu računalnoga modela u obliku sustava GENOBLIK koji je kadar generirati oblike riječi. Uporabne se perspektive takvom sustavu otvaraju i to prije svega s obzirom na »smjer« obrade tj. na izvedbu sustava koji bi bio kadar analizirati oblike riječi. Najjednostavniji bi oblik analize, koji je dovršenjem sustava GENOBLIK postao moguć, moglo biti generiranje svih mogućih oblikâ iz zadanoga leksikona (npr. 30.000 najčešćih riječi) s tim da se uza svaki oblik navede lema i gramatičke kategorije (padež, broj, rod, lice, »vrijeme«, stupanj) dodijeljene tome obliku. Takav bi se popis oblika mogao koristiti za jednostavno uspoređivanje riječi dane na analizu s pohranjenim oblicima. Rezultat pretrage davao bi natuknicu i spisak pridruženih gramatičkih kategorija. Dakako da bi takav postupak prepoznavanja morao dati sva moguća rješenja za slučajeve istopisnica što računalno i nije teško izvedivo.

Takav, međutim, pristup nije racionalan jer bi popis oblika zauzimao otprilike sedamnaest puta više mjesta nego popis osnova. Stoga bi inačica sustava GENOBLIK koja bi mogla pokriti analizu mogla funkcionirati na slijedeći način. Oblik riječi dan na analizu razložio bi se na onoliko načina na koliko bi to bilo moguće imajući u vidu popis nastavaka. Svaki od nastavaka sudjeluje u ograničenome broju uzoraka čime je odmah na

početku smanjen broj osnova iz leksikona koje mogu tvoriti morfotaktički niz analizirane riječi. Nadalje, preoblike ne djeluju na čitavu osnovu već najčešće na jedno od tri pismena s njezina desna kraja. Time se jednostavnim usporedbom početnoga pismena osnove i riječi dane na analizu može još više suziti broj osnova-kandidata za razrješenje analize. Konačno, se nad tako ograničenim skupom osnova mogu u obliku radne hipoteze primijeniti transformacije i provjeriti odgovara li tako preobličena osnova s nastavkom riječi dane na analizu. Rezultata, dakako, može biti više, a može ga i ne biti ukoliko u leksikonu ne postoji osnova koja bi u nekom obliku svoje paradigme bila ista s likom riječi danim na analizu.

Kakve su dobiti od takvoga sustava u obradi korpusa, prije svega u postupcima obilježavanja pojava gramatičkim kategorijama (*tagging*) i(li) lematizaciji, ovdje doista nije potrebno eksplicirati. Rezultat generiranja svih oblika ovakvome sustavu zadanoga leksikona može biti takav popis riječi koji u drugim jezicima, za koje već postoji, čini nezaobilaznu bazu pri izvedbi programa za automatsko otkrivanje grešaka u tekstovima (*spelling checker*).

To su samo neke od perspektiva praktične prirode koje se otvaraju ovako postavljenome sustavu. Za teorijske dorade dakako da uvijek ima mjesta unatoč težnji da se razradi sustav kadar generirati *sve različite* oblike neke riječi. U kojoj se to mjeri doista ovim radom uspjelo izvesti ostavljeno je na daljnju prosudbu.

## 7. LITERATURA

### 7.1. Knjige i članci

Akhmanova, Olga (1971) *Phonology, Mophonology, Morphology*, Mouton, Den Haag-Pariz.

Anderson, Stephen (1988) »Inflection« u Hammond – Noonan (1988), str. 23-44.

Aronoff, Mark (1976) *Word Formation in Generative Grammar*, MIT Press, Cambridge-London.

Babić, Stjepan (1985) »Sročnost s količinskim riječima« u *Jezik* god. 32, br. 3, str. 65-75.

Babić, Stjepan (1986) *Tvorba riječi u hrvatskom književnom jeziku – Nacrt za gramatiku*, JAZU-Globus, Zagreb.

Babić, Stjepan (1989) »Jesu li prilozi promjenljive riječi?« u *Jezik* god. 36, br. 3, str. 84-87.

Babić, Stjepan (1992) »Sintetski prikaz promjene imenica« u *Radovi zavoda za slavensku filologiju* 27, 1992, str. 179-186.

Babić, Stjepan – Finka, Božidar – Moguš, Milan (1994) *Hrvatski pravopis*, Školska knjiga, Zagreb.

Babić, Stjepan – Brozović, Dalibor – Moguš, Milan – Pavešić, Slavko – Škarić, Ivo – Težak, Stjepko (1991) *Povijesni pregled, glasovi i oblici hrvatskoga književnog jezika*, HAZU-Globus, Zagreb. (GRAM)

Babić, Zrinka (1991) *Generativni opis konjugacijskih oblika*, Znanstvena biblioteka HFD-a, Zagreb.



- Barić, Eugenija – Lončarić, Mijo – Malić, Dragica – Pavešić, Slavko – Peti, Mirko – Zečević, Vesna – Znika, Marija (1990) *Gramatika hrvatskoga književnog jezika*, Školska knjiga, Zagreb, (2. izdanje).
- Bear, John (1986) »A Morphological Recognizer with Syntactic and Phonological Rules« u *COLING 1986*, str. 272-276.
- Bear, John (1988) »Morphology with Two-Level Rules and Negative Rule Features« u *COLING 1988*, str. 28-31.
- Becker Makkai, Valerie (1969) On the Correlation of Morphemes and Lexemes u Binnick-Davison-Green-Morgan (1969), str. 159-166.
- Binnick, Robert I. – Davison, Alice – Green, Georgia M. – Lorgan, Jerry L. (ur.) (1969) *Papers from the Fifth Regional Meeting*, April 18-19. 1969, Chicago Linguistic Society, Chicago
- Bjarkman, Peter Christian (1975) »Toward a Proper Conception of Process in Natural Phonology« u Grossman-James San-Vance (1975), str. 60-72.
- Boras, Damir (1990) »Rječnička baza kao osnova za izradu automatskog detektora pogrešaka teksta na hrvatskom jeziku pisanog pomoću kompjutora« u Tkalac-Tuđman (1990), str. 57-74.
- Botha, Rudolf P. (1968) *The Function of the Lexicon in Transformational Generative Grammar*, Mouton, Den Haag-Pariz.
- Brabec, Ivan – Hraste, Mate – Živković, Sreten (1954) *Gramatika hrvatskoga ili srpskoga jezika*, Školska knjiga, Zagreb, (2. izdanje).
- Brozović, Dalibor – Ivić, Pavle (1988) *Jezik, srpskohrvatski/hrvatskosrpski, hrvatski ili srpski*, izvadak iz II. izdanja Enciklopedije Jugoslavije, Leksikografski zavod »Miroslav Krleža«, Zagreb.
- Boguraev, Bran – Briscoe, Ted (ur.) (1989) *Computational Lexicography for Natural Language Processing*, Longman, London-New York.

- Bybee, Joan L. (1988) »Morphology as Lexical Organization« u Hammond – Noonan (1988), str. 119-143.
- Byrd, Roy J. – Calzolari, Nicoletta – Chodorow, Martin S. – Klavans, Judith L. – Neff, Mary S. – Rizk, Omneya A. (1987) »Tools and Methods for Computational Lexicology« u *Computational Linguistics*, Vol. 13, 3-4, str. 219-240.
- Cahill, Lynne J. (1990) »Syllable-based Morphology« u *COLING 1990*, Vol. 3, str. 48-53.
- Carroll, Lewis (1985) *Alica u zemlji čudesa i iza zrcala*, GZH, Zagreb.
- Carstairs, Andrew (1988) »Nonconcatenative Inflection and Paradigm Economy« u Hammond – Noonan (1988), str. 71-78
- Chomsky, Noam (1957, <sup>11</sup>1984) *Sintaksičke strukture*, Dnevnik-Književna zajednica Novog Sada, Novi Sad 1984.
- Chomsky, Noam (1965) *Aspects of the Theory of Syntax*, MIT Press, Cambridge MA.
- Chomsky, Noam (1979) *Gramatika i um*, Nolit, Beograd.
- D'Agostino, F. B. (1976) »The Relevance of Mathematical Linguistics to Empirical Linguistics« u *York Papers in Linguistics*, 6, str. 165-184.
- De Bleser, Ria – Bayer, Josef (1988) »On the Role of Inflectional Morphology in Agrammatism« u Hammond – Noonan (1988), str. 45-70.
- Dembitz, Šandor (1993) *Automatizacija postupka otkrivanja grešaka u tekstu u novim telekomunikacijskim službama*, doktorska disertacija, Elektrotehnički fakultet Sveučilišta u Zagrebu.
- Di Sciullo, Anna-Maria – Williams, Edwin (1987) *On the Definition of the Word*, MIT Press, Cambridge MA.

- Domenig, Marc (1990) »Lexeme-Based Morphology: A Computationally Expensive Approach Intended for a Server-Architecture« *u COLING 1990*, Vol. 2, str. 77-86.
- Dressler, Wolfgang U. (1988) »Preferences vs. Strict Universals in Morphology: Word-Based Rules« *u Hammond – Noonan (1988)*, str. 143-154.
- Dumitrescu, Cristan (1992) »Paradigmatic Morphology Modeling and Lexicon Design with MORPHO-2« *u EURALEX '92 PROCEEDINGS*, Malaga, str. 203-212.
- Erjavec, Tomaž – Tancig, Peter (1987) »Pregled nekaterih računalniških pristopov k morfološki analizi jezika« *u Vitas (1990)*, str. 118-122.
- Erjavec, Tomaž – Tancig, Peter (1988) »Dvo-nivojski model kot teorija in program za morfološko analizo in sintezo« *u ROJP 4*, str. 199-206.
- Finkler, Wolfgang – Neumann, Günter (1988) »MORPHIX, Fast Realization of a Classification-Based Approach to Morphology«, interna publikacija projekta SFB 314 (XTRA), Bericht Nr. 40, Juni 1988.
- Golding, Andrew R. – Thompson, Henry S. (1985) »A Morphology Component for Language Programs« *u Linguistics 23*, str. 263-284.
- Goldsmith, John (ur.) (1993) *The Last Phonological Rule*, University of Chicago Press, Chicago-London.
- Görz, Günther – Paulus, Dietrich (1988) »A Finite State Approach to German Verb Morphology« *u COLING 1988*, str. 212-215.
- Grossman, Robin E. – James San, L. – Vance, Timothy J. (ur.) (1975) *Papers from the Eleventh Regional Meeting*, April 18-20. 1975, Chicago Linguistic Society, Chicago.
- Hajič, Jan (1988) »Formal Morphology« *u COLING 1988*, str. 222-224.
- Halle, Morris (1973) »Prolegomena to a Theory of Word Formation« *u Linguistic Inquiry*, Vol. 4, No. 1, 3-16.

- Hammond, Michael – Noonan, Michael (ur.) (1988) *Theoretical Morphology*, Academic Press, Inc., San Diego-London.
- Hammond, Michael – Noonan, Michael (1988b) »Morphology in the Generative Paradigm« u Hammond – Noonan (1988), str. 1-19.
- Hou, John Y. (1975) »Isn't Rule Ordering Necessary in Phonology« u Grossman-James San-Vance (1975), str. 295-301.
- Jäppinen, Harri – Ylilampi, Matti (1986) »Associative Model of Morphological Analysis: An Empirical Inquiry« u *Computational Linguistics*, Vol. 12, No. 4, str. 257-272.
- Johns, David A. (1969) »On the Correlation of Morphemes and Lexemes« u Binnick-Davison-Green-Morgan (1969), str. 374-381.
- Kačić, Miro (1992) »Neki osnovni lingvistički pojmovi i nazivi« u *Suvremena lingvistika* 34, str. 119-126.
- Karlsson, Fred (ur.) (1985) *Computational Morphosyntax, Report on Research 1981-84*, Publications of the Dept. of General Linguistics, Univ. of Helsinki.
- Karttunen, Lauri (1993) *Finite-State Lexicon Compiler*, Xerox PARC, Palo Alto.
- Karttunen, Lauri (1993) »Finite-State Constraints« u Goldsmith (1993), str. 173-194.
- Karttunen, Lauri – Beesley, Kenneth R. (1992) *Two-Level Rule Compiler*, Xerox PARC, Palo Alto.
- Katičić, Radoslav (1986) *Sintaksa hrvatskoga književnog jezika*, JAZU-Globus, Zagreb.
- Kay, Martin (1977) »Morphological and Syntactic Analysis« u Zampolli (1977), str. 131-233.

- KordiĆ, Snježana (1992) »Determinator — vrsta riječi ili funkcionalni razred?« u *Suvremena lingvistika* 33, str. 27-32.
- Koskenniemi, Kimmo (1983) *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*, Publications of the Dept. of General Linguistics, Univ. of Helsinki.
- Koskenniemi, Kimmo – Church, Kenneth Ward (1988) »Complexity, Two-Level Morphology and Finnish« u *COLING 1988*, str. 335-340.
- Kržak, Miroslav (1988) »Serbo-Croatian Morpho-spelling« u *ROJP* 4, str. 207-214.
- Kržak, Miroslav (1990) »Opisna, stohastička i relacijska gramatika na primjeru morfologije hrvatskog književnog jezika« u *Tkalac-Tuđman* (1990), str. 39-55.
- Kržak, Miroslav – Boras, Damir (1985) »Rječnička baza hrvatskog književnog jezika« u *Informatologia Yugoslavica* 17 (3-4), str. 223-242.
- Lopina, Vjera (1990) »Jezično znanje na primjeru tvorbe riječi« u *Tkalac-Tuđman* (1990), str. 33-38.
- Lopina, Vjera (1992) »Dvorazinski opis morfonoloških smjena u pisanome hrvatskom jeziku« u *Suvremena lingvistika* 34, str. 185-194.
- Lopina, Vjera (1993) »Dvorazinski model morfološkoga opisa« u *Tkalac-Tuđman* (1993), str. 75-80.
- Maretić, Tomo (1899) *Gramatika i stilistika hrvatskoga ili srpskoga književnog jezika*, Naklada knjižare L. Hartmana (Kugli & Deutsch), Zagreb.
- Mihaljević, Milan (1991) *Generativna i leksička fonologija*, Školska knjiga, Zagreb.
- Moguš, Milan (1993) *Povijest hrvatskoga jezika*, Globus, Zagreb.

- Molbæk Hansen, Peter (1989) »Syntax, Morphology, and Phonology in Text-to-Speech Systems« u *ARIPUC (Annual Report of the Institute of Phonetics, University of Copenhagen)*, vol. 23, str. 119-152.
- Pala, Karel (1990) »An Algorithmic Dictionary of Czech for the IBM PC (and compatibles)«, izlaganje na *Conference on Computational Lexicography*, Balatonfüred, 8. 9.-11. 9. 1990.
- Pala, Karel – Osolsobě, Klára (1990) »Czech stem dictionary for PC XT/AT«, izlaganje na *Conference on Computational Lexicography*, Balatonfüred, 8. 9.-11. 9. 1990.
- Pavešić, Slavko – Težak, Stjepko – Babić, Stjepan: »Oblici hrvatskoga književnog jezika (morfologija)« u Babić, Stjepan – Brozović, Dalibor – Moguš, Milan – Pavešić, Slavko – Škarić, Ivo – Težak, Stjepko: *Povijesni pregled, glasovi i oblici hrvatskoga književnog jezika*, HAZU-Globus, Zagreb 1991, str. 460-741.
- Pavlović-Lazetić, Gordana – Wong, Eugene (1986) »Managing Text as Data« u *Proceedings of the Twelfth International Conference on Very Large Data Bases*, Kyoto, August 1986.
- Pavlović-Lazetić, Gordana (1987) »Morfološka analiza vođena pravilima« u *Vitas* (1990), str. 136-141.
- Perlmutter, David M. (1988) »The Split Morphology Hypothesis: Evidence from Yiddish« u Hammond – Noonan (1988), str. 79-100.
- Pounder, Amanda – Kommenda, Markus (1986) »Morphological Analysis for German Text-to-speech System« u *COLING 1986*, str. 263-268.
- Pranjković, Ivo (1982) »Klasifikacija riječi na vrste i razine gramatičke analize« u *Radovi Zavoda za slavensku filologiju* 17, 1982, str. 23-30 str.
- Pranjković, Ivo (1992) »Prilozi kao riječi 'sviju vrsta'« u *Suvremena lingvistika* 34, str. 243-249.

- Rakić, Stanimir (1985) »O raspodjeli krajnjeg zvučnog suglasnika u srpskohrvatskom jeziku« u *Jezik* 33, broj 2, str. 33-64.
- Řehák, Vladimír (1968) *Od hipotetičkog do egzaktnog u jeziku*, Školska knjiga, Zagreb.
- ROJP 4, Zbornik radova 4. konferencije *Računalniška obdelava jezikovnih podatkov*, Portorož, 3. 10.-7. 10. 1988.
- Russel, G. J. – Ritchie, G. D. – Pulman, S. G. – Black, A. W. (1986) »A Dictionary and Morphological Analyser for English« u *COLING 1986*, str. 277-279.
- Samardžija, Marko (1988) »Duga i kratka množina u hrvatskom književnom jeziku« u *Jezik*, god. 35, br. 5, str. 129-136.
- Schmidtbauer, Paulina Rocha (1972) »Pokušaj opisa imeničke deklinacije hrvatskoga jezika« u *Suvremena lingvistika* 5-6, str. 49-64.
- Silić, Josip (1992) »Status skupova *št* i *žd* u hrvatskom jeziku« u *Suvremena lingvistika* 34, str. 263-280.
- Simov, Kiril – Angelova, Galia – Paskaleva, Elena (1990) »MORPHO-ASSISTANT: The Proper Treatment of Morphological Knowledge« u *COLING 1990*, Vol. 3., str. 455-457.
- Srhoj-Čerina, Ljubica (1986) »Kolebanja u dugoj množini« u *Jezik*, god. 33, br. 5, str. 148-150.
- Stemberger, Joseph Paul – MacWhinney, Brian (1988) »Are Inflected Forms Stored in Lexicon?« u Hammond – Noonan (1988), str. 101-116
- Škiljan, Dubravko (1980) *Pogled u lingvistiku*, Školska knjiga, Zagreb.
- Tadić, Marko (1992) *Računalna obrada morfologije hrvatskoga književnog jezika na imeničnom potkorpusu*, magistarski rad, Sveučilište u Zagrebu.
- Tadić, Marko (1992b) »Problemi računalne obrade imeničnih oblika u hrvatskome« u *Suvremena lingvistika* 34, str. 301-307.

- Tadić, Marko (1992c) »Od korpusa do čestotnoga rječnika hrvatskoga književnog jezika« u *Radovi Zavoda za slavensku filologiju* 27, 1992, str. 169-178.
- Tafra, Branka (1988) »Razgraničavanje opisnih i odnosnih pridjeva« u *Rasprave Zavoda za jezik*, sv. 14, str. 185-197.
- Tafra, Branka (1989) »Što su brojevi?« u *Rasprave Zavoda za jezik*, sv. 15, str. 219-237.
- Tkalac, Slavko – Tuđman, Miroslav (ur.) (1990) *Informacijske znanosti i znanje*, Zavod za informacijske studije, Zagreb.
- Tkalac, Slavko – Tuđman, Miroslav (ur.) (1993) *Obrada jezika i prikaz znanja*, Zavod za informacijske studije, Zagreb.
- Trost, Harald (1990) »The Application of Two-Level Morphology to Non-Concatenative German Morphology« u *COLING 1990*, Vol. 2, str. 371-376.
- Tzoukermann, Evelyne – Liberman, Mark Y. (1990) »A Finite- -State Morphological Processor for Spanish« u *COLING 1990*, Vol. 3, str. 277-282.
- Velčić, Mirna (1987) *Uvod u lingvistiku teksta*, Školska knjiga, Zagreb.
- Vinja, Vojmir (1986) *Jadranska fauna. Etimologija i struktura naziva. Vol. I-II*, JAZU-Logos, Zagreb-Split.
- Vitas, Duško (ur.) (1990) *Matematička i računarska lingvistika: teorija i praksa*, Društvo za primenjenu lingvistiku Srbije, Beograd.
- Vukušić, Stjepan (1991) »Sklonidba imenica vuk, vrag, rog, bog u kratkoj množini« u *Jezik*, god. 38, br. 3, str. 70-72.
- Winograd, Terry (1983) *Language as a Cognitive Process. Vol 1: Syntax*, Addison-Wesley, Reading-Menlo Park-London-Amsterdam-Don Mills-Sydney.



Wothke, Klaus (1986) »Machine Learning of Morphological Rules by Generalization and Analogy« u *COLING 1986*, str. 289-293.

Zampolli, Antonio (1977) *Linguistic Structures Processing*, North-Holland Publishing Co, Amsterdam-New York-Oxford.

## **7.2. Rječnici**

Anić, Vladimir (1991) *Rječnik hrvatskoga jezika*, Novi Liber, Zagreb.

Anić, Vladimir – Silić, Josip (1986) *Pravopisni priručnik hrvatskoga ili srpskoga jezika*, Sveučilišna naklada »Liber« – Školska knjiga, Zagreb.

Babić, Stjepan – Finka, Božidar – Moguš, Milan (1990) *Hrvatski pravopis*, Školska knjiga, Zagreb, (2. izdanje – pretisak).

Benešić, Julije (1990) *Rječnik hrvatskoga književnoga jezika od Preporoda do I. G. Kovačića*, sv. 1-12, HAZU-Globus, Zagreb.

Deanović, Mirko – Jernej, Josip (1963) *Hrvatskosrpsko-talijanski rječnik*, Školska knjiga, Zagreb, (2. izdanje).

Klaić, Bratoljub (1982) *Rječnik, stranih riječi*, Nakladni zavod Matice hrvatske, Zagreb.

Matešić, Josip (1965) *Rückläufiges Wörterbuch des Serbokroatischen*, Band 1, Lieferung 1-4, Otto Harrassowitz, Wiesbaden.

Šugar, Ivan (1990) *Botanički leksikon*, Globus-JAZU, Zagreb.

Šulek, Bogoslav (1990) *Hrvatsko-njemačko-talijanski rječnik znanstvenog nazivlja*, Globus, Zagreb, (pretisak).

Vidović, Radovan (1984) *Pomorski rječnik*, Logos, Split.